Imperial College London

COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Adaptive Signal Processing and Machine Intelligence

Yang Zhao (CID: 01561245)
yang.zhao18@imperial.ac.uk

April 12, 2019

Contents

1 Classical and Modern Spectrum Estimation

1.1 Properties of Power Spectral Density (PSD)

Start from Definition 2 of PSD:

$$P(\omega) = \lim_{N\to\infty} \mathbb{E}\left\{\frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-jn\omega}\right|^2\right\} \tag{1.1}$$

$$= \lim_{N\to\infty} \mathbb{E}\left\{\frac{1}{N}\sum_{m=0}^{N-1} x(m)e^{-jm\omega}\sum_{n=0}^{N-1} x^*(n)e^{jn\omega}\right\} \tag{1.2}$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \mathbb{E}\{x(m)x^*(n)\}\, e^{-j(m-n)\omega} \tag{1.3}$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} r(m-n)e^{-j\omega(m-n)} \tag{1.4}$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{k=-(N-1)}^{N-1} (N-|k|)r(k)e^{-j\omega k} \tag{1.5}$$

$$= \lim_{N\to\infty}\left(\sum_{k=-(N-1)}^{N-1} r(k)e^{-j\omega k} - \frac{1}{N}\sum_{k=-(N-1)}^{N-1} |k|\, r(k)e^{-j\omega k}\right) \tag{1.6}$$

$$= \sum_{k=-\infty}^{\infty} r(k)e^{-j\omega k} - \lim_{N\to\infty}\frac{1}{N}\sum_{k=-(N-1)}^{N-1} |k|\, r(k)e^{-j\omega k} \tag{1.7}$$

With a mild assumption that:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=-(N-1)}^{N-1} |k||r(k)| = 0 \tag{1.8}$$

Equation 1.7 reduces to Definition 1:

$$P(\omega) = \sum_{k=-\infty}^{\infty} r(k)e^{-j\omega k} \tag{1.9}$$

The equality proved the Wiener-Khinchin theorem that if the covariance sequence decays rapidly *(i.e. $r(k)$ is absolutely integrable)*, the autocorrelation function (ACF) and PSD are Fourier transform pairs. An implication is that the energy spectral density (ESD) of the input and output of an LTI system are related by the squared magnitude of the system frequency response:

$$S_{YY}(\omega) = S_{XX}(\omega)|H(\omega)|^2 \tag{1.10}$$

Figure 1 illustrates the PSD of sinusoids and impulse by direct and indirect methods. It can be observed that the covariance sequence $r(k)$ of impulse is still an impulse, so Definition 1 and 2 coincide.

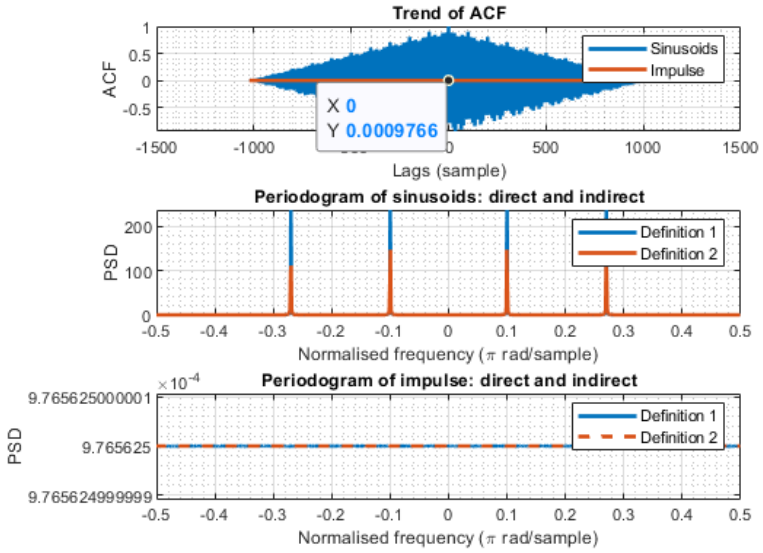

Figure 1: Trend of ACF; PSD of sinusoids and impulse by Definition 1 and 2

On the other hand, the ACF of sinusoids reduces slowly, and Equation 1.8 does not hold. Therefore, the PSD as discrete-time Fourier transform (DTFT) of ACF and average power over frequencies are different in amplitude.

1.2 Periodogram-based Methods Applied to Real–World Data

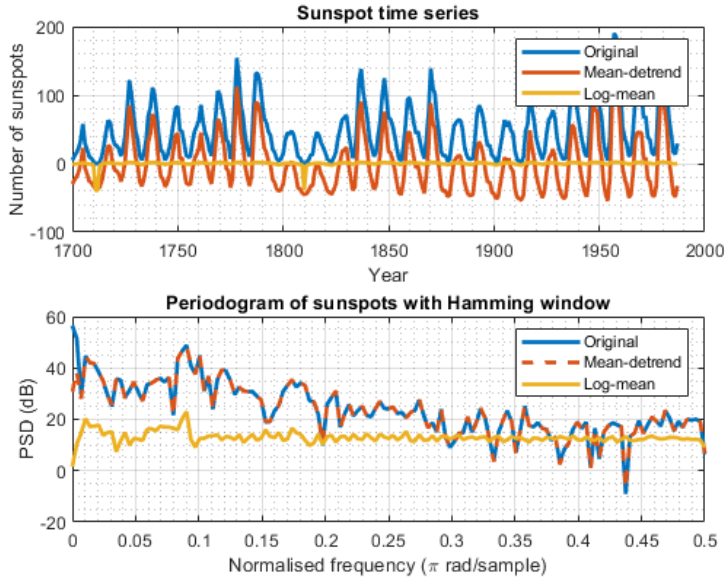

Figure 2: Preprocessed and original sunspot time series; periodogram with Hamming window

(a) Figure 2 suggests that removing the mean and trend can be interpreted as centring and smoothing data in the time domain. It reduces the estimation of DC component below a normalised frequency of 3×10^{-3}, which is related to the average signal value. Also, the detrend function removes the

linear trend from the signal for FFT processing, and the signal experiences less jitter with a denoised spectrum. The influence of preprocessing on the high-frequency component is negligible. In comparison, applying logarithm then subtracting mean provides a stable signal with even fewer jitters. The waveform oscillates significantly when the original signal experiences less periodicity. For instance, an unexpected rise occurs at normalised frequency of 0.1, and the log-mean processed signal successfully captures the variation.

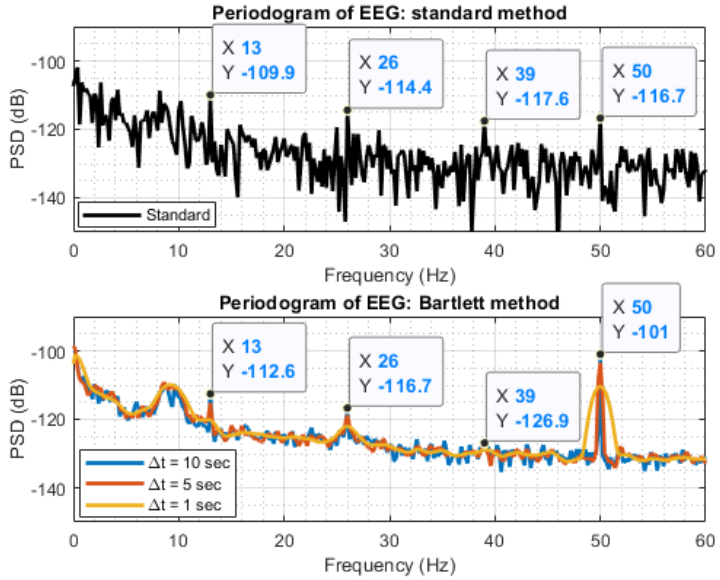

Figure 3: Standard and averaged periodogram of EEG with different window length

(b) We compare the standard periodogram and the averaged periodogram with different window length. Figure 3 shows that both methods managed to detect the SSVEP frequency at 13 Hz and its harmonics. The broad peak around 8 Hz comes from the tiredness of the subject, while the power-line interference at 50 Hz results in the largest magnitude and overwhelms the harmonic at 52 Hz. When window length $\Delta t = 10$, the averaged periodogram has a smaller variance than the standard one, but the dominant peaks are still observable. It suppresses the influence of noise and accentuates the main peaks, which is beneficial to observation. As Δt decreases to 1 second, the periodogram becomes smoother, but the expected peak at 39 Hz becomes hard to recognise. The reason is that using a short window reduces the number of samples in each observation and leads to a worse frequency resolution. In conclusion, there is a trade-off between precision and distinguishability, and a proper window length as 5 second helps to reduce the noise while highlighting the peaks of interest.

1.3 Correlation Estimation

(a) Bias measures the difference between the statistics obtained by a particular sample and the true statistics of the signal. Unbiased estimator ensures fixed mean and no systematic error in the variance obtained by different samples. Figure 4 shows that the autocorrelation of biased and unbiased estimator are similar at small lags $|k| < 100$, but as k increases, the unbiased ACF grows while biased ACF reduces to 0. It comes from the definition:

$$r(k) = \begin{cases} \frac{1}{N} \sum_{n=k+1}^{N} x(n)x^*(n-k) & \text{,biased} \\ \frac{1}{N-k} \sum_{n=k+1}^{N} x(n)x^*(n-k) & \text{,unbiased} \end{cases} \tag{1.11}$$

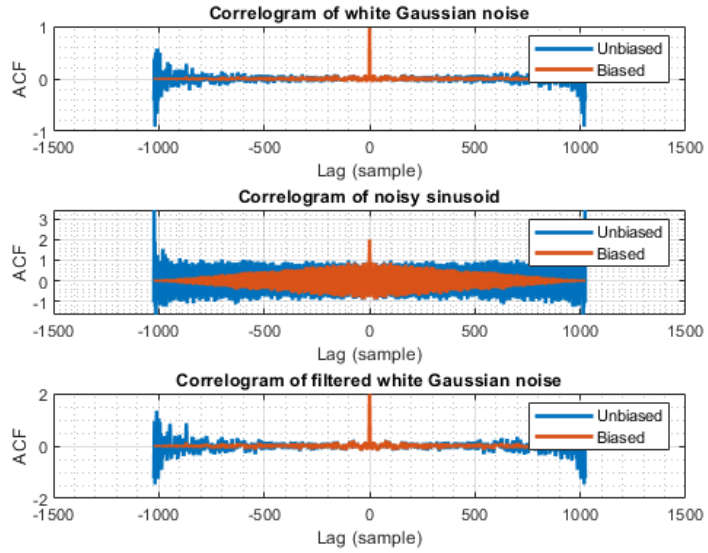

Figure 4: Correlogram of WGN, noisy sinusoid and filtered WGN

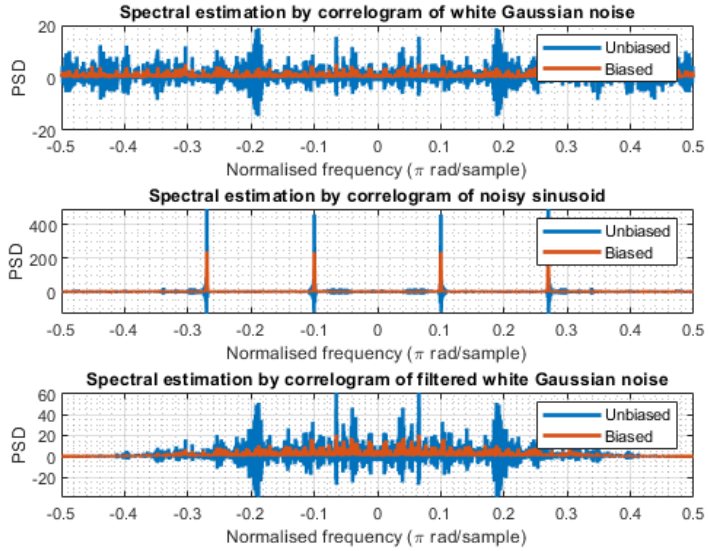

Figure 5: Spectral estimation by correlogram of WGN, noisy sinusoid and filtered WGN

Equation 1.11 suggests that although a large k leads to less similarity and smaller sum, it can have a significant impact on the denominator and produce a large unbiased ACF. Moreover, Figure 5 proves that the PSD by biased estimator agrees with the expectation *(i.e. the spectrum is flat for WGN, impulses for sinusoids, and rectangular for low-pass WGN)*. In contrast, unbiased estimator provides erratic predictions. The reason is that there are fewer samples available to estimate the PSD when k is large, which may ruin the positive definiteness of the ACF and lead to negative meaningless PSD. Therefore, we prefer the biased estimator because it has lower variance and is asymptotically unbiased when N is large.

(b) We generated and analysed 100 different realisations composed of sinusoids and additional WGN (AWGN) of unit variance:

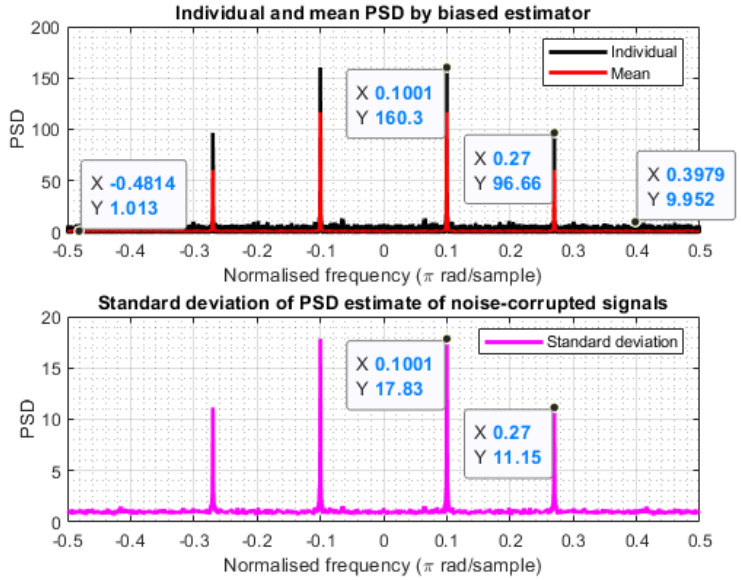

Figure 6: Individual and mean PSD by biased estimator; standard deviation of PSD estimate of noise-corrupted signals

$$x(n) = 0.7\sin(2\pi 0.1n) + 0.5\sin(2\pi 0.27n) + \eta(n) \tag{1.12}$$

Figure 6 shows that although the PSD of AWGN fluctuate between 0 and 10 for different instances, the mean PSD of AWGN has uniform unit power for all frequency components. Although both sinusoidal frequencies are detected successfully, a particular AWGN may add constructively at the frequency of interest. The second plot suggests that standard deviation is proportional to PSD, leading to a large confidence interval when PSD experiences peaks. Moreover, the biased ACF estimator is not consistent because as the sample size increases, the mean does not necessarily converge and the variance approaches the PSD rather than converges to 0.

(c) Figure 7 presents the result in dB scale. With all trends maintained, it zooms in to the small-scale variations and cares less about the fluctuations greater than 1. In other words, the dB plot is an effective presentation that provides more detail in the trends of interest and suppresses the meaningless transitions.

(d) Figure 8 demonstrates that the resolution of periodogram Δf is proportional to the reciprocal of the number of samples $1/N$. For instance, the complex exponentials with a normalised frequency of 0.3 and 0.32 require a minimum sample length $N = 0.89/\Delta f = 44.5$ to be separated clearly [1]. As the number of sample increases, there are more frequency kernels available due to the one-to-one mapping between time and frequency domain. Therefore, the periodogram becomes sharper and closer to the ideal pulses. Note that complex exponentials are half of the sinusoids and not symmetric.

(e) Relevant code is attached below.

```
[X,R] = corrmtx(x,14,'mod');
[S,F] = pmusic(R,2,[ ],1,'corr');
plot(F,S,'linewidth',2); set(gca,'xlim',[0.25 0.40]);
grid on; xlabel('Hz'); ylabel('Pseudospectrum');
```

The first line uses `corrmtx` function to obtain the autocorrelation matrix estimate. Here the input `x` is the noisy exponential, `14` specifies dimension of the correlation matrix is 15-by-15, `'mod'` cor-

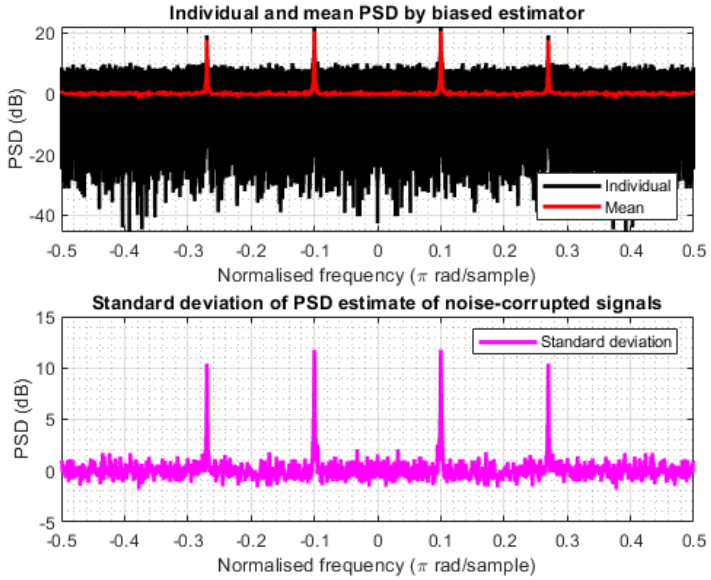

Figure 7: PSD in dB scale

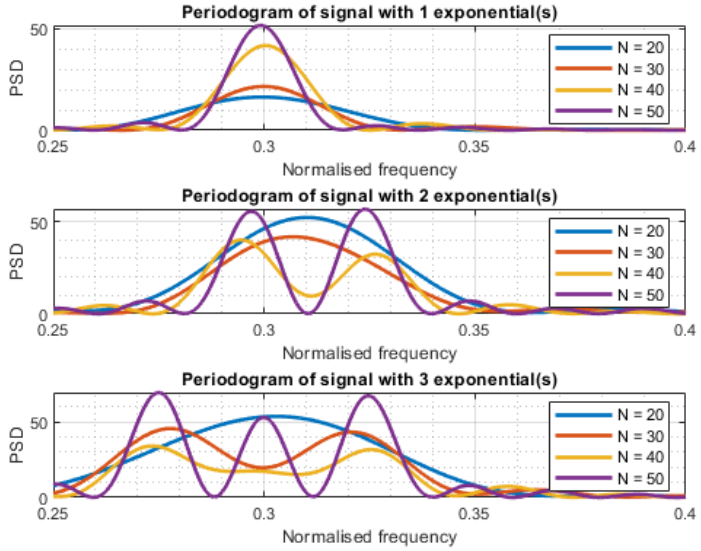

Figure 8: Periodogram of complex exponential signals

responds to the modified covariance method that may produce more accurate estimates. Output `X` is a rectangular Toeplitz matrix, and `X'X` is a biased estimate of the autocorrelation matrix. `R` is the autocorrelation matrix. The second line performs spectrum estimation by multiple signal classification (MUSIC) algorithm. Input `R` is the correlation matrix, `2` is the signal subspace dimensionality, `[ ]` uses the default FFT points, `1` denotes normalised frequency, `'corr'` suggests `R` is the correlation matrix estimate. The output `S` is the pseudospectrum and `f` is normalised frequency. The third line plot the pseudospectrum versus the normalised frequency and limit the frequency range to [0.25, 0.40].

Figure 9 shows that the standard deviation of MUSIC estimate experience peaks at the frequencies of complex exponentials, similar to the case of periodogram. It also suggests that the MUSIC algorithm

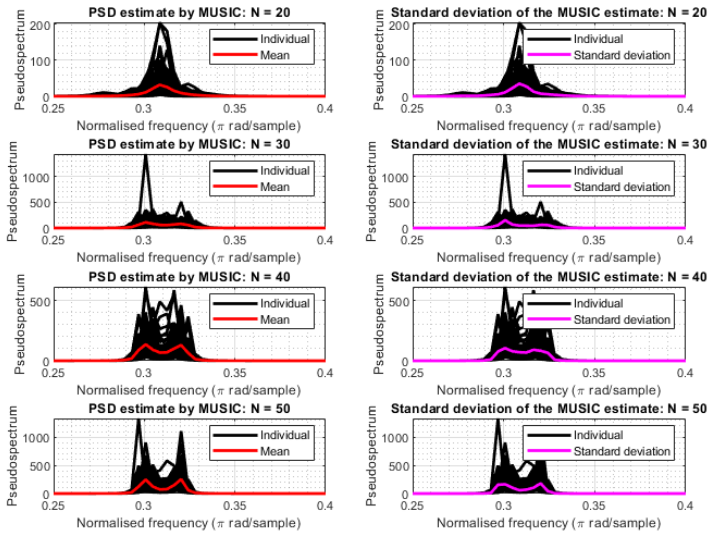

Figure 9: Spectral estimation of complex exponentials by MUSIC algorithm

cannot distinguish frequency components of 0.3 and 0.32 with only 20 samples. As N increases to 30, the two peaks are captured although the trend is not very obvious. In comparison, the red curves in the middle plot of Figure 8 by periodogram cannot separate the closely spaced components with 30 samples. It proves that the MUSIC algorithm provides more detailed information and outperforms the periodogram when only a few samples are available, which is a superresolution approach because the estimation function can be evaluated at any frequency.

Although the MUSIC algorithm performs better in identifying adjacent frequency components for a limited sample, it is a subspace method that requires the dimension p as the input. In this instance it is assumed that $p = 15$, but for a general MUSIC estimation, the signal subspace dimension should be determined by algorithms as [2] and [3]. A large p increases the complexity, while a small p may reduce the accuracy due to loss of information.

1.4 Spectrum of Autoregressive (AR) Processes

(a) AR parameter $\mathbf{a}$ comes from the Yule-Walker equation:

$$\mathbf{r}_{xx} = \mathbf{R}_{xx}\mathbf{a} \Rightarrow \mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx} \tag{1.13}$$

where $\mathbf{R}_{xx}$ is the ACF matrix. The biased estimator provides a positive definite Toeplitz $\mathbf{R}_{xx}$ that is invertible. However, the invertibility is not guaranteed for unbiased estimators, and AR parameter $\mathbf{a}$ may not be available by Equation 1.13.

(b) Figure 10 illustrates that low order models with large prediction error cannot describe the variations properly. As order increases, there can be more freedom in fitting the sample, and the prediction error always decreases. However, it increases system complexity and may lead to overfitting. Although the original signal is of order 4, the biased estimator suggests using a model with order 7 for a trade-off between error and complexity. Also, the result on different samples can be different, due to the bias and the small sample size. Hence, the unbiased estimator is not a proper alternative, and the model can be improved by increasing the sample size.

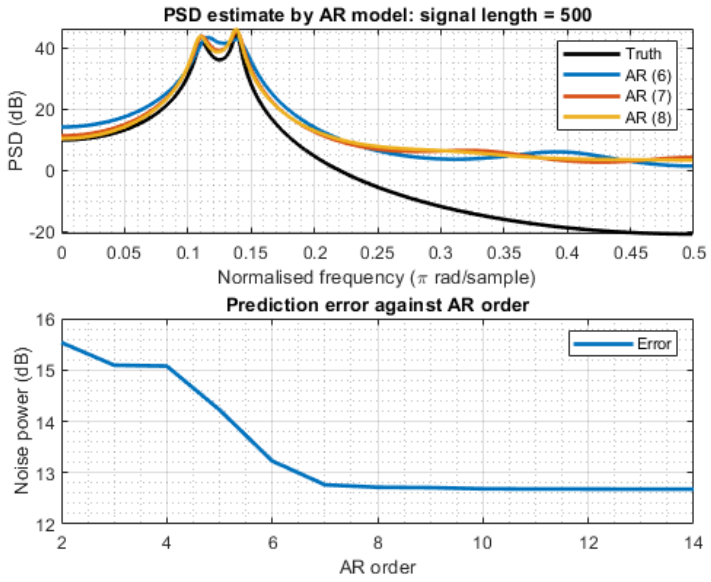

Figure 10: PSD estimate by AR model for signal of length 500; prediction error against AR order

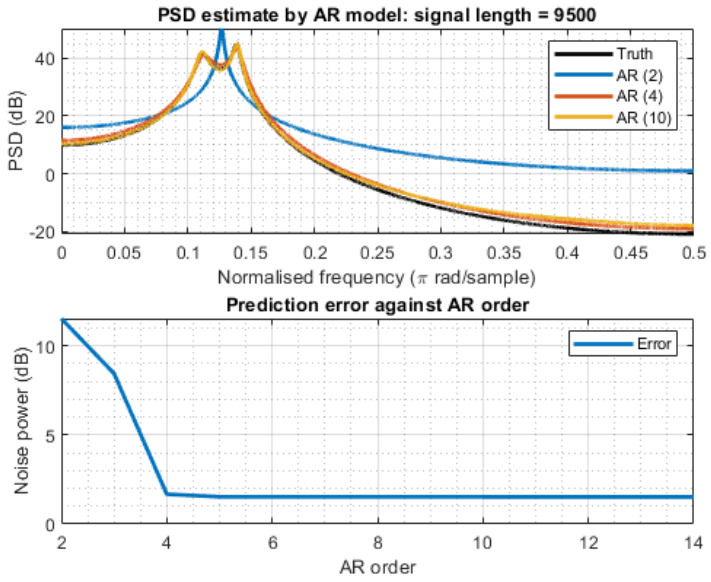

Figure 11: PSD estimate by AR model for signal of length 9500; prediction error against AR order

(c) Figure 11 demonstrates that large sample size help to reduce the influence of randomness on model selection. In other words, the sample provides more information about the AR process and behaves more like the hidden pattern. As expected, the signal model is successfully estimated as AR (4) with sufficient samples. AR models with order lower than 4 cannot capture the variations of interest, while high-order models tend to overfit the noise with high complexity.

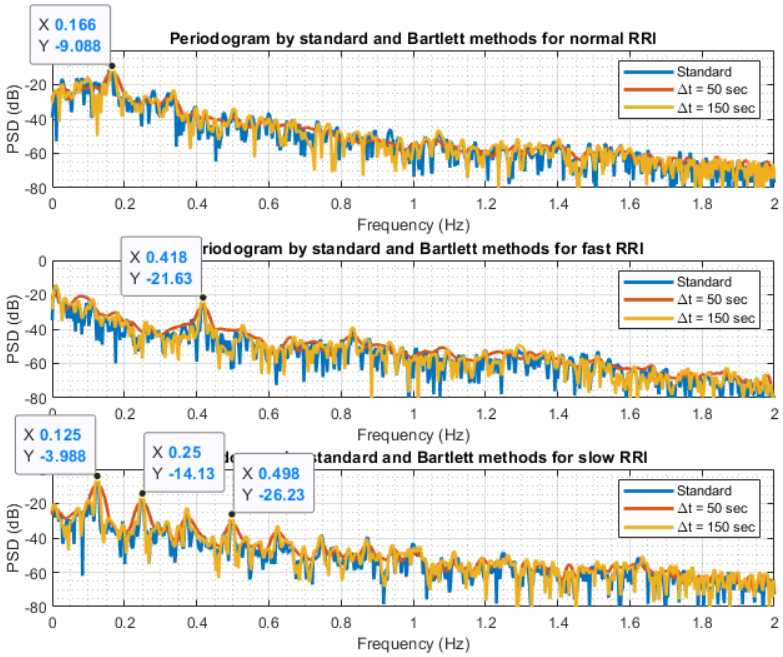

Figure 12: Standard and averaged periodogram for RRI data

1.5 Real World Signals: Respiratory Sinus Arrhythmia from RRI

(a) We compared the periodogram of RRI data by standard and Bartlett methods with window lengths $\Delta t = 50, 150$ second. The sample is divided into individual segments, and Hamming window without overlap is applied to each part. The periodograms are averaged to reduce the influence of noise. Result is shown in Figure 12.

(b) There is a trade-off between frequency resolution and noise reduction. As window length decreases from 240 (standard) to 150 seconds, the amplitude of fluctuation is significantly reduced while the details are maintained. In contrast, the periodogram corresponding to $\Delta t = 50$ only shows the trend (envelope) with broad peaks, but the variance is much smaller. Also, using a smaller window with lower resolution results in more evident harmonics. It suggests that averaging over subsamples is helpful to reduce noise but results in lower frequency resolution. Therefore, the window length should be carefully chosen for the balance in between.

Discard the harmonics, the peaks of interest occurs at $f_n = 0.166$, $f_f = 0.418$, $f_s = 0.125$ Hz. The relationship between frequency and breaths per minute (BPM) is $BPM = 2 \times 60 \times f$, which suggests BPM = 20, 50, 15 for normal, fast, slow respirations. Both the standard and averaged approaches ensure valid results.

(c) Figure 13 contrasts the periodogram and AR estimates of RRI data. To reproduce the peaks of interest, the optimal AR model order for normal, fast, slow respirations are 2, 10, 6 respectively. Compared with the periodogram method, the optimal AR spectrum estimate only describes the overall trend of the observation and ignore the influence of noise and harmonics, which is simple and effective to highlight the main characteristics. However, improper order p can lead to under or over modelling, and modelling secondary features *(e.g. harmonics)* require much higher order and needs larger memory.

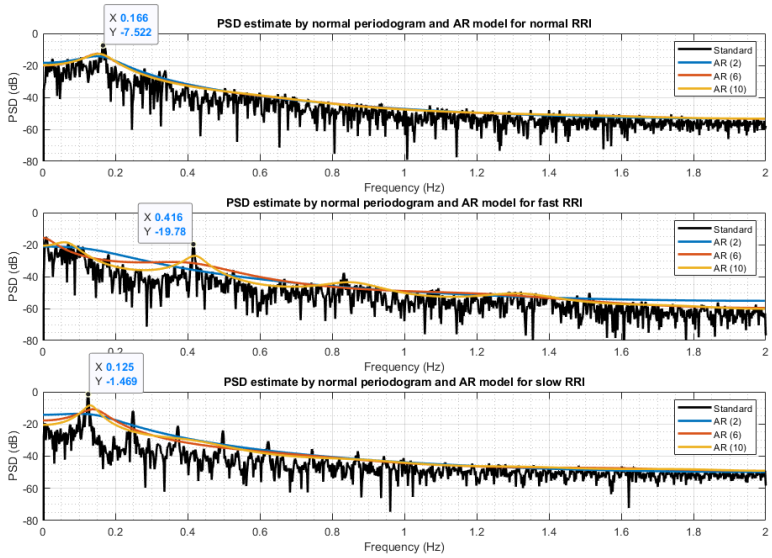

Figure 13: PSD estimate by normal periodogram and AR model for RRI data

1.6 Robust Regression

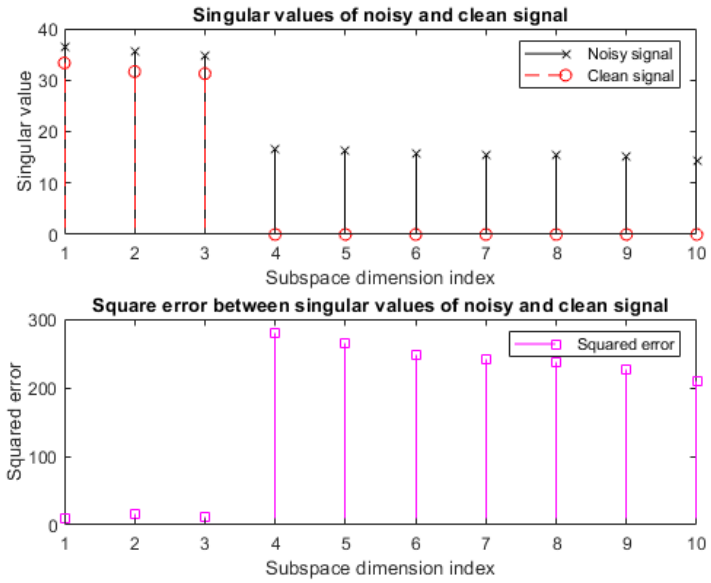

Figure 14: Singular values and squared error of clean and noisy signals

(a) Figure 14 shows that the clean signal $\mathbf{X}$ has a rank of 3, while the noisy signal $\mathbf{X}_{noise}$ is rank-10 with 3 dominant singular values. The reason is that the signal only has non-zero components in the signal subspace, while noise always occupies the whole space. The introduction of noise increases the singular value slightly in dimension 1 to 3, but the boost is more evident in dimension 4 to 10. It leads to a small error in the signal subspace and large deviation in other irrelevant dimensions. Hence, it is possible to determine the signal subspace dimension by singular value decomposition. However, if the signal-to-noise ratio (SNR) is small, the error magnitude will be similar in all dimensions, and it will be hard to identify the rank of $\mathbf{X}_{noise}$.

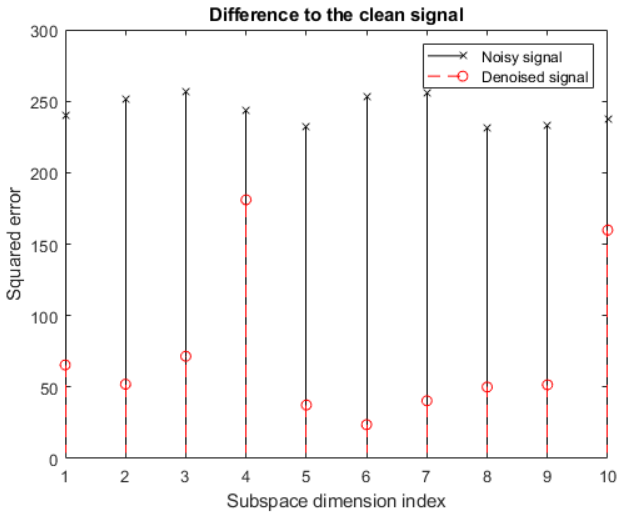

Figure 15: Difference to the clean signal

(b) The denoised signal $\tilde{\mathbf{X}}_{noise}$ is a low-rank approximation of $\mathbf{X}_{noise}$ with $r = 3$ most significant components. Figure 15 suggests that the denoised signal is a better approximation of the original signal, which demonstrates that the most significant information is stored in a few principal components. An interesting phenomenon is that the error between $\mathbf{X}$ and $\tilde{\mathbf{X}}_{noise}$ is different among dimensions. One possible reason is that the low-rank approximation overfits the noise in $\mathbf{X}_{noise}$ and amplifies the magnitude difference between dimensions.

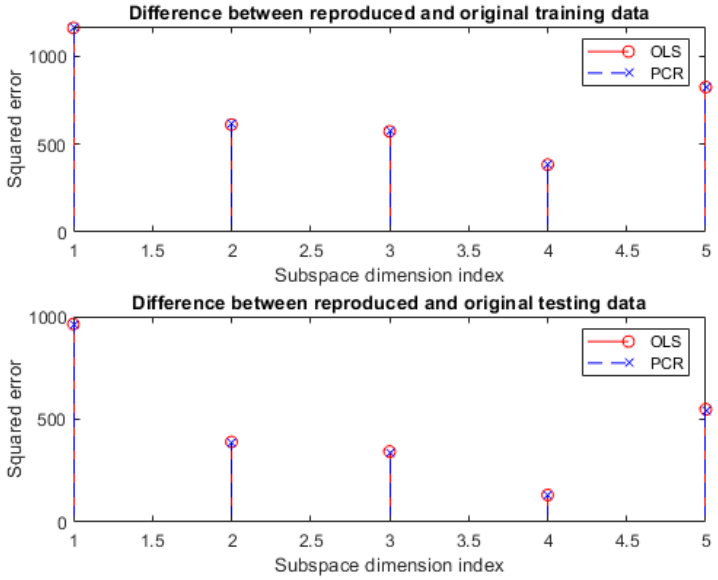

Figure 16: Estimation error by OLS and PCR

(c) The ordinary least squares (OLS) and principal component regression (PCR) solutions are given by:

$$\hat{\mathbf{B}}_{OLS} = \left(\mathbf{X}_{noise}^T \mathbf{X}_{noise}\right)^{-1} \mathbf{X}_{noise}^T \mathbf{Y} \tag{1.14}$$

Table 1: Summation of Prediction Error Square on training and testing data

	Training	Testing
OLS	3551.7	2377.4
PCR	3577.1	2353.9

$$\hat{\mathbf{B}}_{PCR} = \mathbf{V}_{1:r}(\Sigma_{1:r})^{-1}\mathbf{U}_{1:r}^{T}\mathbf{Y} \tag{1.15}$$

Figure 16 shows the estimation error on training and testing data by OLS and PCA methods using $r = 3$ most significant components. The summation of error square is provided by Table 1. It appears that OLS fits the training set better, while PCR provides a smaller error on the testing data.

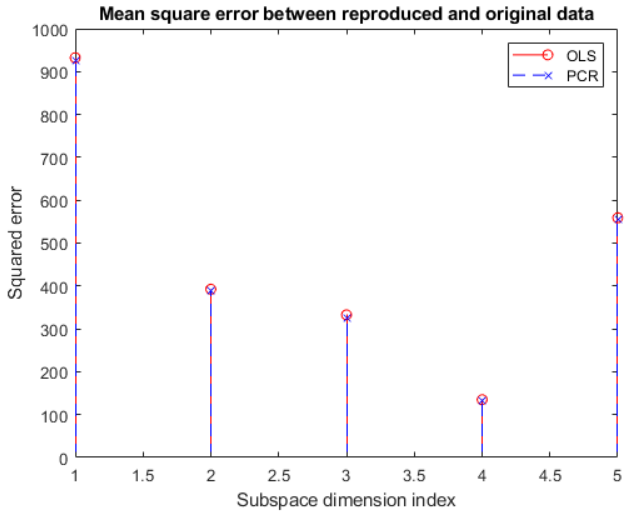

Figure 17: MSE over 100 testing sets

(d) OLS and PCR are tested over 100 generated sets, and the mean square error (MSE) on each dimension is illustrated by Figure 17. The MSE sum up to 2348.2 for OLS and 2338.7 for PCR. In conclusion, both algorithms provide acceptable results, and PCR outperforms OLS with a 0.4% smaller MSE.

2 Adaptive signal processing

2.1 The Least Mean Square (LMS) Algorithm

(a) The correlation matrix $\mathbf{R}_{xx}$ of the input vector $\mathbf{x}(n) = [x(n-1), x(n-2)]^T$ writes:

$$\mathbf{R}_{xx} = \mathbb{E}\left\{\mathbf{x}(n)\mathbf{x}(n)^T\right\} \tag{2.1}$$

$$= \mathbb{E}\left\{ \begin{array}{cc} x(n-1)^2 & x(n-1)x(n-2) \\ x(n-2)x(n-1) & x(n-2)^2 \end{array} \right\} \tag{2.2}$$

$$= \left(\begin{array}{cc} r_{xx}(0) & r_{xx}(1) \\ r_{xx}(-1) & r_{xx}(0) \end{array} \right) \tag{2.3}$$

where $r_{xx}(k)$ denotes ACF of $\mathbf{x}(n)$:

$$
\begin{aligned}
r_{xx}(k) &= \mathbb{E}\{x(n)x(n-k)\} && (2.4)\\
&= \mathbb{E}\{[a_1 x(n-1) + a_2 x(n-2) + \eta_n]x(n-k)\} && (2.5)\\
&= a_1 \mathbb{E}\{x(n-1)x(n-k)\} + a_2 \mathbb{E}\{x(n-2)x(n-k)\} + \mathbb{E}\{\eta(n)x(n-k)\} && (2.6)\\
&= \begin{cases} a_1 r_{xx}(1) + a_2 r_{xx}(2) + \sigma_\eta, & k = 0 \\ a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2), & k \neq 0 \end{cases} && (2.7)
\end{aligned}
$$

Therefore, ACF $r_{xx}(k)$ with $k = 0, 1, 2$ reduces to:

$$
\begin{cases} r_{xx}(0) = a_1 r_{xx}(1) + a_2 r_{xx}(2) + \sigma_\eta \\ r_{xx}(1) = a_1 r_{xx}(0) + a_2 r_{xx}(-1) \\ r_{xx}(2) = a_1 r_{xx}(1) + a_2 r_{xx}(0) \end{cases} \tag{2.8}
$$

Note $r_{xx}(-1) = r_{xx}(1)$ due to symmetry. Solving Equation 2.8 with $a_1 = 0.1, a_2 = 0.8, \sigma_\eta^2 = 0.25$ gives $r_{xx}(0) = 0.9259, r_{xx}(1) = 0.4630$. Therefore, the correlation matrix is:

$$
\mathbf{R}_{xx} = \begin{pmatrix} 0.9259 & 0.4630 \\ 0.4630 & 0.9259 \end{pmatrix} \tag{2.9}
$$

Eigendecomposition gives $\lambda_1 = 0.4630$, $\lambda_2 = 1.3889$. According to [1], the LMS algorithm converges in the mean if the step size μ satisfy:

$$
0 < \mu < \frac{2}{\lambda_{\max}} = \frac{2}{1.3889} = 1.44 \tag{2.10}
$$

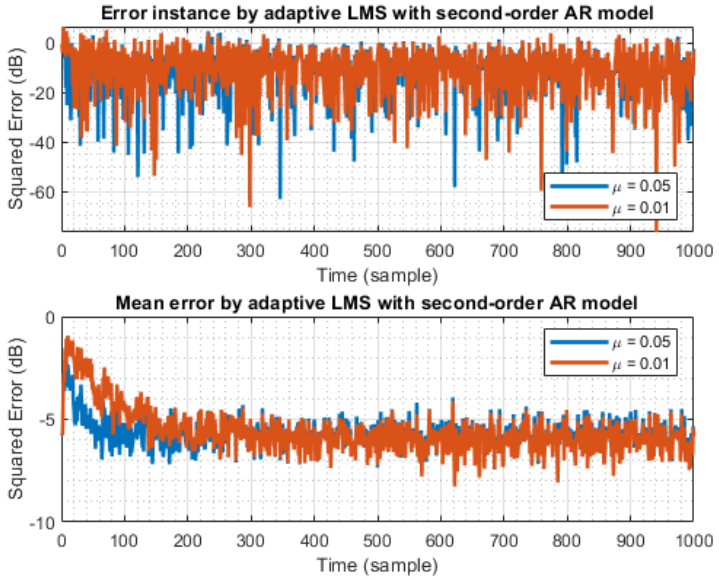

Figure 18: Error of adaptive LMS with different learning rate

Table 2: Accurate and approximated misadjustments

μ	EMSE	$\mathcal{M}$	$\mathcal{M}_{approx}$
0.05	0.0123	0.0493	0.0463
0.01	0.0025	0.0101	0.0093

(b) Figure 18 illustrates the typical and mean prediction error over 100 realisations by adaptive LMS with $N = 1000$ samples. It can be observed from the learning curves that a large step size of $\mu = 0.05$ ensures fast converge within 50 samples, while it takes 150 trials for $\mu = 0.01$ to converge. Also, the red curve is slightly below the blue curve, suggesting a smaller steady state error. It is because a larger learning rate leads to steeper steps in gradient descent, requiring fewer steps to reach the minimum error point. In the steady state, iteration will continue around the optimum point, and the error depends on the step size.

(c) Define the excess mean square error (EMSE) to measure the difference of adaptive filter MSE and minimum Wiener filter MSE, the misadjustment $\mathcal{M} = EMSE/\sigma_\eta^2$ can be approximated as $\mathcal{M} \approx (\mu/2)Tr(\mathbf{R})$ for small step sizes. Table 2 based on 100 realisations demonstrates that the approximation error is 6.1% for $\mu = 0.05$ and 8.1% for $\mu = 0.01$, which is reasonably accurate and acceptable. However, the approximation should have been more accurate for smaller step size. One possible reason is that the duration of the transient state to discard is under-estimated.

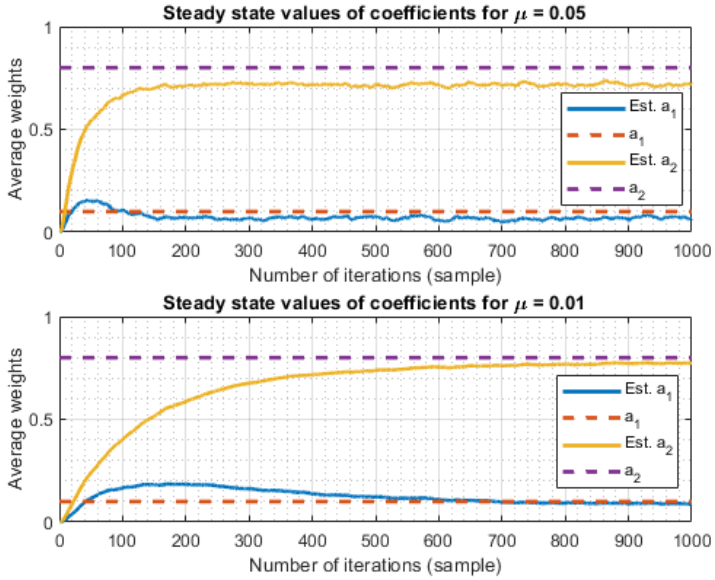

Figure 19: Steady state values of adaptive LMS filter coefficients for different step sizes

(d) Figure 19 indicates the update of average coefficients over 100 realisations. As discussed in (b), large step size $\mu = 0.05$ leads to fast converge within 140 iterations but large steady-state jitter up to 0.1 for $\hat{a}_1$ and 0.025 for $\hat{a}_2$. In comparison, it takes 600 steps for the filter with $\mu = 0.01$ to converge, and the coefficient difference in the steady state is less than 0.03 for $\hat{a}_1$ and 0.01 for $\hat{a}_2$. There is a trade-off between converge speed and steady-state error. A large step size ensures fast gradient descent, and it takes a short time to approach the minimum error point. However, it results in a large overshoot, and the coefficients will oscillate around.

(e) Begin from the cost function $J_2(n)$:

$$J_2(n) = \frac{1}{2}\left(e^2(n) + \gamma \|\mathbf{w}(n)\|_2^2\right) \tag{2.11}$$

$$= \frac{1}{2}\left(\left(y(n) - \mathbf{w}(n)^T\mathbf{x}(n)\right)^T\left(y(n) - \mathbf{w}(n)^T\mathbf{x}(n)\right) + \gamma\mathbf{w}(n)^T\mathbf{w}(n)\right) \tag{2.12}$$

Its gradient w.r.t. weight $\mathbf{w}$ is:

$$\nabla_{\mathbf{w}} J_2(n) = -\left(y(n) - \mathbf{w}(n)^T\mathbf{x}(n)\right)\mathbf{x}(n) + \gamma\mathbf{w}(n) \tag{2.13}$$

$$= -e(n)\mathbf{x}(n) + \gamma\mathbf{w}(n) \tag{2.14}$$

Update weight by the steepest descent (SD) method:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(-\nabla_{\mathbf{w}} J_2(n)) \tag{2.15}$$

$$= \mathbf{w}(n) + \mu(e(n)\mathbf{x}(n) - \gamma\mathbf{w}(n)) \tag{2.16}$$

$$= (1 - \mu\gamma)\mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{2.17}$$

The result corresponds to the expression of leaky LMS:

$$\mathbf{w}(n+1) = (1 - \mu\gamma)\mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{2.18}$$

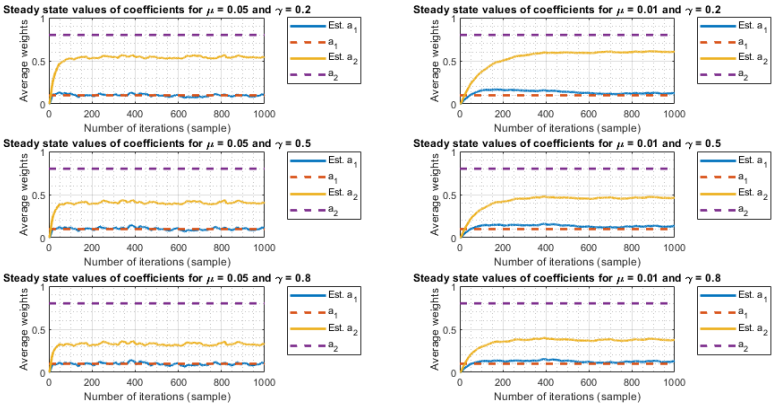

Figure 20: Steady state values of adaptive leaky LMS filter coefficients for different step sizes and leakages

(f) Figure 20 illustrates the coefficients of leaky LMS. As mentioned above, a large step size leads to fast convergence and great oscillation in the steady state. Interestingly, for a non-zero leakage γ, the filter coefficients do not converge to the ground truth. As γ increases, the gap becomes wider, and the steady-state jitter is amplified. It is because the adaptive filter approaches the optimum Wiener filter:

$$\mathbf{w}^\star = \mathbf{R}^{-1}\mathbf{p} \tag{2.19}$$

The optimum solution requires the invertibility of the autocorrelation matrix $\mathbf{R}$ that is not guaranteed. In comparison, the leaky LMS converges to:

$$\mathbf{w}^\star_{leaky} = (\mathbf{R} + \gamma\mathbf{I})^{-1}\mathbf{p} \tag{2.20}$$

An introduction of leakage γ ensures the invertibility and the existence of $\mathbf{w}^{\star}_{leaky}$. However, the deviation is proportional to γ. An interpretation is that the leakage denotes the degree of forgettability. The filter updates the weight based on prediction error and the previous weight, and information will be lost if there is a leakage.

2.2 Adaptive Step Sizes

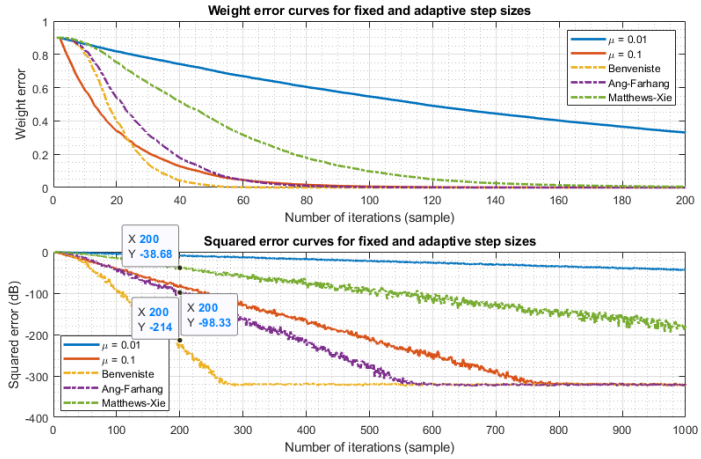

Figure 21: Weight error and error square curves for fixed and adaptive step sizes with $\mu_0 = 0$

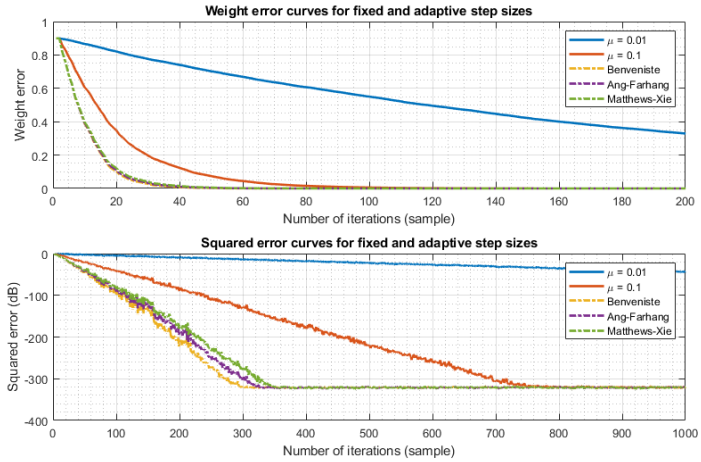

Figure 22: Weight error and error square curves for fixed and adaptive step sizes with $\mu_0 = 0.2$

(a) With a fixed step size, the standard LMS has a relatively low complexity of $\mathcal{O}(1)$. Nevertheless, its main disadvantage is the trade-off between the convergence speed and steady-state error variance. The gradient adaptive step size (GASS) adjust μ based on the gradient of the cost function. Large step size is used in the beginning for fast convergence, and it is reduced as the estimated parameters become closer to the truth.

Figure 21 contrasts the weight error and learning curves of LMS with fixed and adaptive step sizes by Benveniste, Ang-Farhang and Matthews-Xie algorithms. Notice that the initial step size $\mu_0 = 0$ and the scale $\rho = 5 \times 10^{-3}$ for all GASS algorithm, and we set $\alpha = 0.8$ for Ang-Farhang. It can be observed that Benveniste updates the step size frequently and converges faster than the others. The

advantage comes from the $\mathbf{x}(n-1)\mathbf{x}(n-1)^T$ term with complexity $\mathcal{O}(M^2)$ where M is the model order. Ang-Farhang reaches the steady state slightly faster than $\mu = 0.1$ within 80 iterations with complexity $\mathcal{O}(M)$. However, Matthews-Xie with $\mathcal{O}(M)$ updates the step size slower than the others, and the performance is worse than $\mu = 0.1$. Moreover, the steady-state error after 200 iterations is -214 dB for Benveniste, -98 dB for Ang-Farhang, -39 dB for Matthews-Xie, -80 dB for $\mu = 0.1$, -10 dB for $\mu = 0.01$. The convergence speed of three GASS algorithms can be optimised further by selecting a proper initial step size as $\mu_0 = 0.2$, but the range should obey Equation 2.10. An instance is given in Figure 22. In conclusion, Benveniste outperforms Ang-Farhang and Matthews-Xie with higher complexity, and with a suitable initial step size μ_0, GASS results in faster convergence and smaller steady-state variance than the standard LMS.

(b) The weight update equation is given by:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_p(n)\mathbf{x}(n) \tag{2.21}$$

Hence, the posteriori error $e_p(n)$ writes

$$\begin{aligned} e_p(n) &= d(n) - \mathbf{x}^T(n)\mathbf{w}(n+1) && (2.22)\\ &= d(n) - \mathbf{x}^T(n)\Big(\mathbf{w}(n) + \mu e_p(n)\mathbf{x}(n)\Big) && (2.23)\\ &= d(n) - \mathbf{x}^T(n)\mathbf{w}(n) - \mu e_p(n)\|\mathbf{x}(n)\|^2 && (2.24)\\ &= e(n) - \mu e_p(n)\|\mathbf{x}(n)\|^2 && (2.25)\\ &= \frac{e(n)}{1+\mu\|\mathbf{x}(n)\|^2} && (2.26) \end{aligned}$$

Therefore, the weight update $\Delta\mathbf{w}(n)$ is:

$$\begin{aligned} \Delta\mathbf{w}(n) &= \mathbf{w}(n+1) - \mathbf{w}(n) && (2.27)\\ &= \mu e_p(n)\mathbf{x}(n) && (2.28)\\ &= \mu\frac{e(n)}{1+\mu\|\mathbf{x}(n)\|^2}\mathbf{x}(n) && (2.29)\\ &= \frac{e(n)}{\frac{1}{\mu}+\|\mathbf{x}(n)\|^2}\mathbf{x}(n) && (2.30) \end{aligned}$$

In comparison, the standard normalised LMS (NLMS) reads:

$$\Delta\mathbf{w}(n) = \frac{\beta e(n)}{\varepsilon + \|\mathbf{x}(n)\|^2}\mathbf{x}(n) \tag{2.31}$$

It suggests that the update equation based on a posteriori error $e_p(n)$ boils down to the NLMS algorithm with $\beta = 1, \varepsilon = 1/\mu$.

(c) Generalised normalised gradient descent (GNGD) is an extension of NLMS, where the regularisation factor ε is time-varying:

$$\varepsilon(n+1) = \varepsilon(n) - \rho\mu\frac{e(n)e(n-1)\mathbf{x}^T(n)\mathbf{x}(n-1)}{\Big(\varepsilon(n-1)+\|\mathbf{x}(n-1)\|^2\Big)^2} \tag{2.32}$$

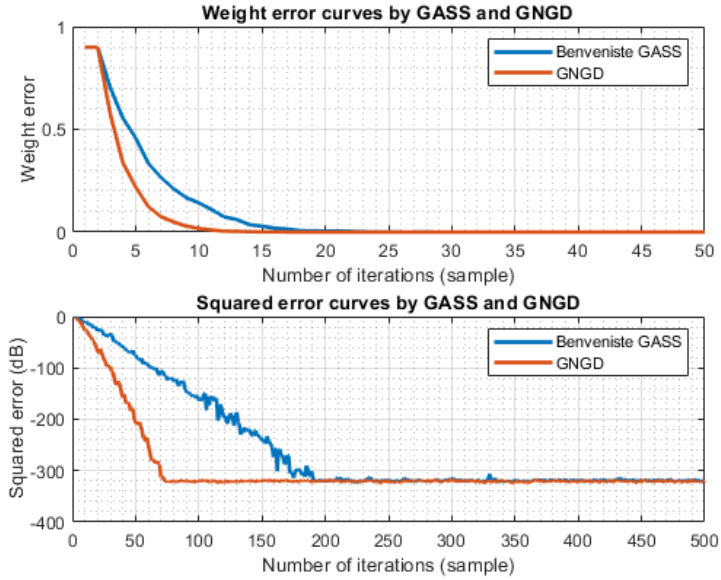

Figure 23: Weight error and error square curves by Benveniste GASS ($\mu_0 = 0.5$) and GNGD $\mu_0 = 7$

Figure 23 contrasts the performance of Benveniste GASS and GNGD. Notice that the scale $\rho = 5{\times}10^{-3}$, and the optimum initial step size is found by trials as $\mu_0 = 0.5$ for Benveniste GASS and $\mu_0 = 7$ for GNGD. It can be observed that the GNGD converges within 10 iterations while the Benveniste GASS reaches stability after 17 updates. Also, the steady-state error of GNGD is significantly smaller than Benveniste GASS with fewer oscillations. The reason is that GNGD utilises an additional adaptive factor to provide non-linear adjustment and stabilise the NLMS, which is more suitable for nonlinear and nonstationary signals and more robust to the initialisation [4]. The complexity of GNGD is $\mathcal{O}(M)$ as there is no outer product in Equation 2.32. Compared with Benveniste GASS with $\mathcal{O}(M^2)$, GNGD guarantees faster convergence and smaller steady-state oscillation with much lower complexity.

2.3 Adaptive Noise Cancellation

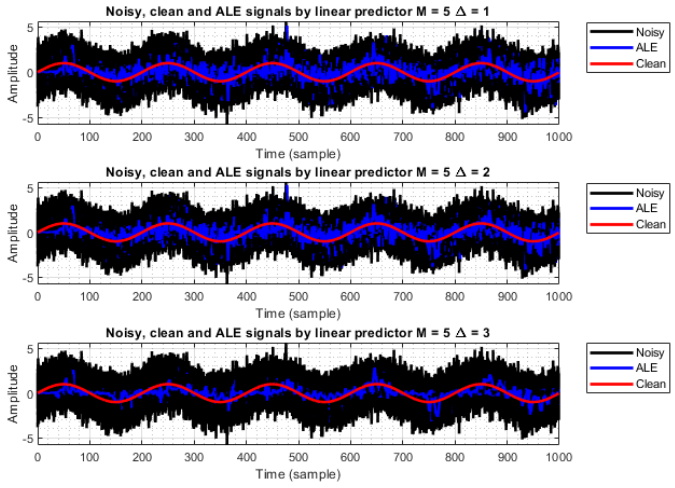

Figure 24: Noisy, clean and ALE signals by linear predictor with different delays

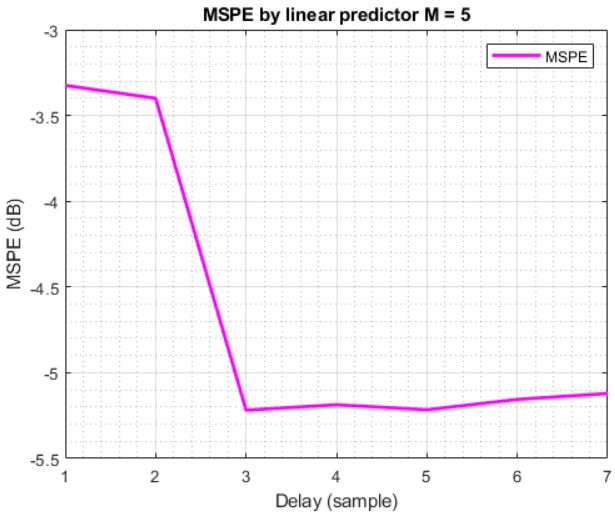

Figure 25: MSPE by linear predictor with different delays

(a) The MSE can be expressed as:

$$\mathbb{E}\left\{(s(n)-\hat{x}(n))^2\right\} = \mathbb{E}\left\{(x(n)+\eta(n)-\hat{x}(n))^2\right\} \tag{2.33}$$

$$= \mathbb{E}\left\{(x(n)-\hat{x}(n))^2\right\} + \mathbb{E}\left\{\eta^2(n)\right\} + 2\mathbb{E}\left\{(x(n)-\hat{x}(n))\,\eta(n)\right\} \tag{2.34}$$

$$= \mathbb{E}\left\{(x(n)-\hat{x}(n))^2\right\} + \mathbb{E}\left\{\eta^2(n)\right\} + 2\mathbb{E}\left\{x(n)\eta(n)\right\} - 2\mathbb{E}\left\{\hat{x}(n)\eta(n)\right\} \tag{2.35}$$

where $\eta(n) = v(n) + 0.5v(n-2)$ is coloured noise, $v(n)$ is white noise with unit variance, and $\hat{x}(n) = \mathbf{w}^T(n)\mathbf{u}(n)$ is enhanced signal. The first term $\mathbb{E}\left\{(x(n)-\hat{x}(n))^2\right\}$ is irrelevant to noise $\eta(n)$ for a fixed model. The second term $\mathbb{E}\left\{\eta^2(n)\right\}$ measures the noise power which is also a constant. The third term $2\mathbb{E}\{x(n)\eta(n)\}$ is 0 because signal and noise are uncorrelated and the noise has a zero mean. Therefore, we only need to consider the last term $-2\mathbb{E}\{\hat{x}(n)\eta(n)\}$ to minimise the MSE and guarantee the adaptability. It writes:

$$\mathbb{E}\{\hat{x}_\Delta(n)\eta(n)\} = \mathbb{E}\left\{\mathbf{w}^T(n)\mathbf{u}_\Delta(n)\eta(n)\right\} \tag{2.36}$$

$$= \mathbb{E}\left\{\sum_{k=0}^{M-1} w_k s(n-\Delta-k)\eta(n)\right\} \tag{2.37}$$

$$= \mathbb{E}\left\{\sum_{k=0}^{M-1} w_k \left(x(n-\Delta-k)+\eta(n-\Delta-k)\right)\eta(n)\right\} \tag{2.38}$$

$$= \mathbb{E}\left\{\sum_{k=0}^{M-1} w_k \eta(n-\Delta-k)\eta(n)\right\} \tag{2.39}$$

$$= \mathbb{E}\left\{\sum_{k=0}^{M-1} w_k \left(v(n-\Delta-k)+0.5v(n-\Delta-k-2)\right)\left(v(n)+0.5v(n-2)\right)\right\} \tag{2.40}$$

Since $\mathbb{E}\{v(n-i)v(n-j)\} = 0$ iff $i \neq j$ for i.i.d. white noise, a minimum delay $\Delta_{\min} = 3$ can ensure no overlapping between $v(n-\Delta-k)+0.5v(n-\Delta-k-2)$ and $v(n)+0.5v(n-2)$, and $\mathbb{E}\{\hat{x}_\Delta(n)\eta(n)\} = 0$ to minimise the MSE.

Figure 24 and 25 demonstrates the conclusion above. It can be observed that a delay no smaller than 3 can reproduce the signal with MSPE lower than -5 dB. However, as Δ increases, the autocorrelation of the signal is reduced and the MSPE is increased.

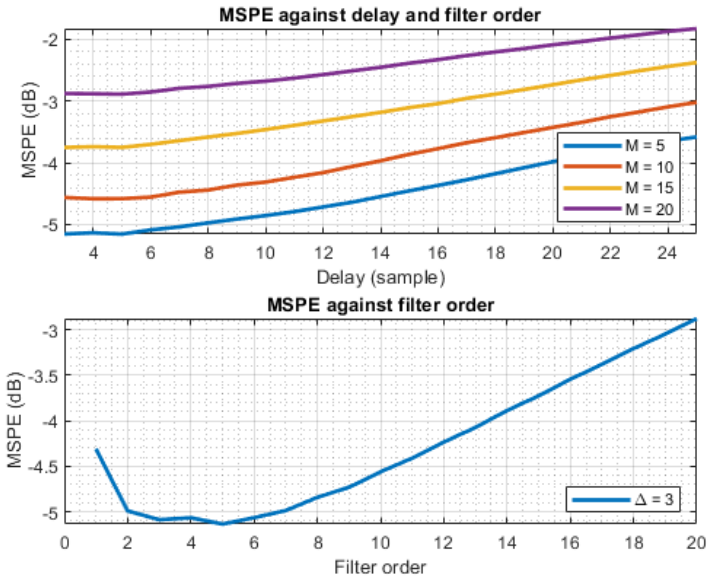

Figure 26: MPSE, delay and filter order

(b) Figure 26 shows the relationship between MSPE, delay and filter order. MSPE is flat for $\Delta \leq 5$ and grows steadily as delay increases. It is because adaptive line enhancer (ALE) relies on the assumption that the noise is uncorrelated with its delayed version while the signal has a strong autocorrelation. When the delay is large, the autocorrelation of the desired signal is reduced and the MSPE is increased. The result also suggests that the filter order M should be at least 2 to capture the variations successfully. Interestingly, in the environment of coloured noise, increasing filter order leads to a larger MSPE for $M > 5$. The reason is that the linear predictor tends to over-model the coloured noise as part of the signal pattern. As the computational cost increases with filter order, it is recommended to use $\Delta = 3$ and $M = 2$ or 3 in this case to balance MSPE and complexity.

(c) Figure 27 proves the advantage of adaptive noise cancellation (ANC) over ALE with a secondary noise correlated to the coloured noise by $\varepsilon(n) = 0.9\eta(n) + 0.05$. The MSPE in the steady state is -5.94 dB for ALE and -10.13 dB for ANC. It takes 300 iterations for ALE to find the right sinusoid pattern with small steady-state variance. In comparison, the improvement over time is not obvious for ALE. Therefore, ANC is preferred when there are sufficient samples, and ALE can be an alternative when the sample size is small.

(d) ALE can be extended to remove particular frequency component by first introducing noise of corresponding frequency then applying ALE. Figure 28 illustrates the spectrogram of preprocessed POz using a window length of 4096 and an overlap ratio of 0.9. There is a strong 50 Hz component by the power line, and we set the reference input as a 50 Hz sine wave with unit amplitude corrupted by the white noise of variance 0.01.

Figure 29 shows the influence of filter order M and step size μ on the denoising performance. It can be observed that for a fixed step size, increasing order leads to a cleaner result but the white noise component may be removed if M is too large. On the contrary, under-modelling can happen if M is small, and there can be some remaining parts as denoted by the lower left plot. Therefore, the order determines the degree of denoising.

The effect of varying μ can be observed by vertical contrast. A large μ covers a wider frequency range

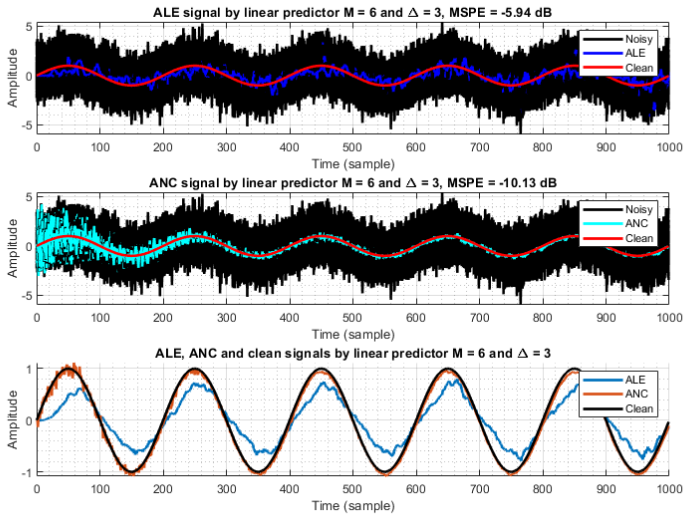

Figure 27: ALE and ANC signals by linear predictor M = 6 and Δ = 3

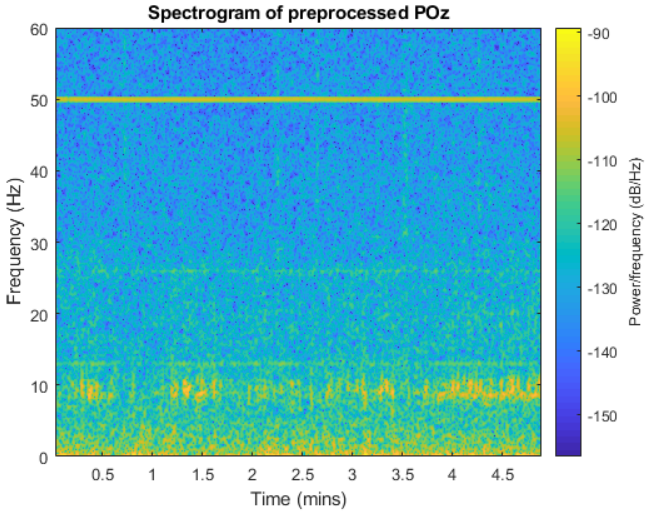

Figure 28: Spectrogram of preprocessed POz

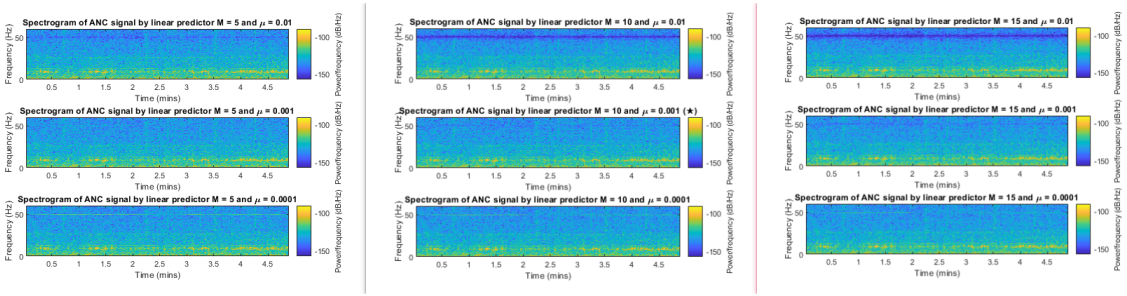

Figure 29: Spectrogram of ANC signal by linear predictor with different order and step size

and may affect the neighbouring frequency components. However, the 50 Hz component has a width of 2 Hz and is not removed thoroughly for all $\mu = 1 \times 10^{-4}$ cases. To suppress the 50 Hz noise and maintain the information stored in other frequencies, an optimal setting of $M = 10, \mu = 1 \times 10^{-3}$ is

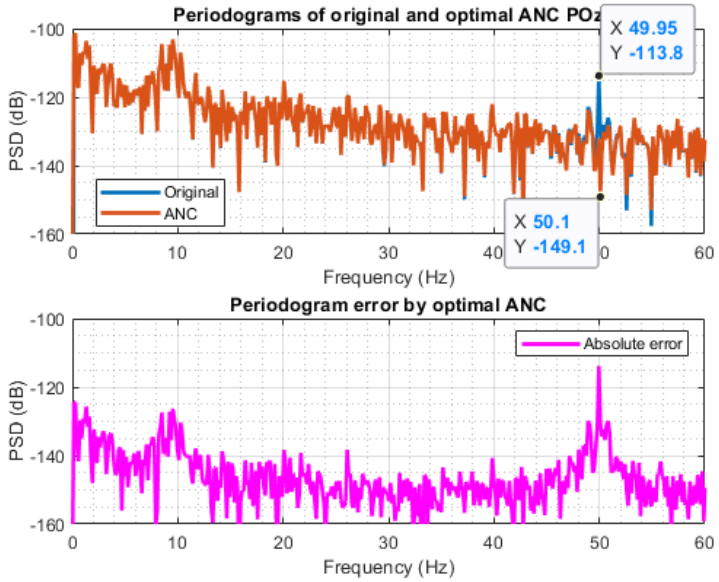

Figure 30: Periodograms of original and optimal ANC POz; periodogram error by optimal ANC

employed.

Figure 30 presents the periodograms before and after ANC processing. As expected, both curves almost overlap for all frequency besides 50 Hz, where a difference of 35.3 dB exists between the original and ANC signals.

3 Widely Linear Filtering and Adaptive Spectrum Estimation

3.1 Complex LMS and Widely Linear Modelling

(a) The complex LMS (CLMS) and augmented CLMS (ACLMS) are modified on LMS for the estimation of complex-valued signals. Both algorithms are tested on a first-order widely-linear MA (WLMA). Figure 31 indicates that the white noise is circularly distributed with $\rho = 0.13$, whose expectation is 0 but the deviation comes from the sample size $N = 1000$ being not large enough. The WLMA (1) is with elliptical distribution with $\rho = 0.86$. Moreover, the learning curves suggest that CLMS is not suitable for the WLMA (1) process with nearly constant squared error = 8.9 dB, as the degree of freedom in CLMS is 1 but the elliptical distribution of WLMA (1) requires 2. In contrast, ACLMS models the process well with the help of the augmented weight $\mathbf{g}(n)$ by considering $\mathbf{x}(n)$ and its conjugate, and the result converges after 450 steps with a steady-state error -306.2 dB. To conclude, ACLMS is more general and suitable for the signals with non-circular distribution, while CLMS is computationally less expensive for circular signals.

(b) Figure 32, 33, 34 show the scatter diagram of measured and predicted distribution for low, medium, high wind regimes. As the circularity coefficient measures the degree of non-circularity, $\rho_{low} = 0.16, \rho_{medium} = 0.45, \rho_{high} = 0.62$ imply the circularity of the data decreases as wind speed increases. Therefore, CLMS is more suitable for low wind regime while ACLMS is more proper for medium and high wind regimes, as denoted by the figures. Moreover, increasing the order M leads to more scattered distributions and there can be some outliers with extremely large amplitude for ACLMS due to overfitting.

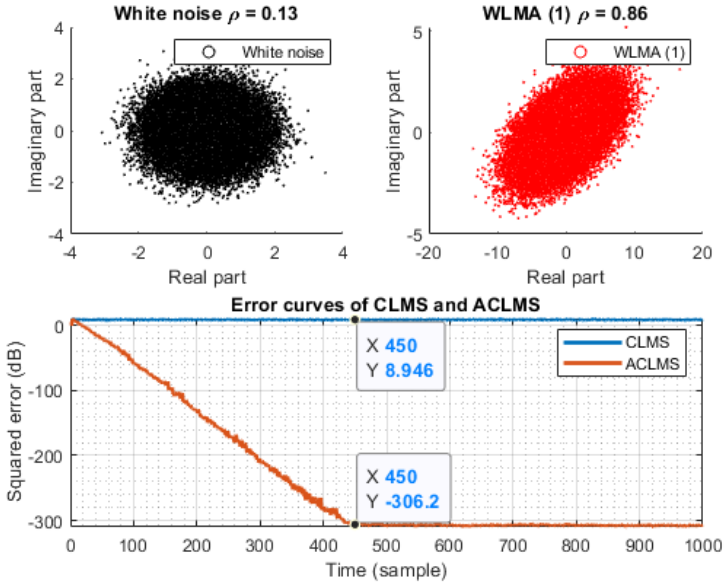

Figure 31: Distribution of white noise and WLMA (1); error curves of CLMS and ACLMS

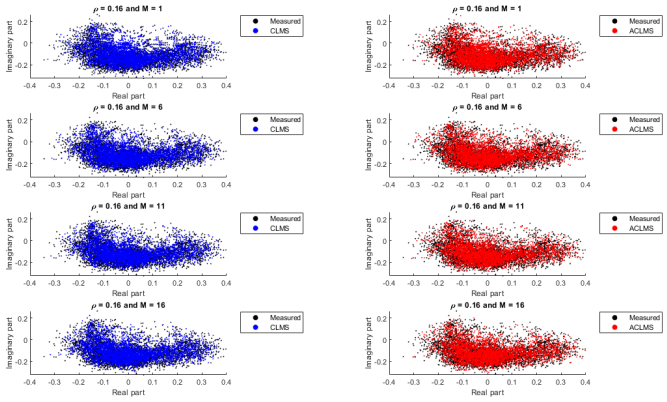

Figure 32: Scatter diagram of low wind regime

Figure 35 presents the relationship between filter order and MSPE. When the order is relatively low *(i.e.* $M < 9, 4, 20$ *for low, medium, high regions)*, ACLMS provides a smaller error than CLMS, because the extra degree of freedom compensates the under-modelling effectively. The advantage is more obvious for high wind regime because of the highly non-circular distribution. In stark contrast, the patterns of low and medium winds are less complicated and more circular, and over-modelling can happen even if M is small. To conclude, the extra parameter $\mathbf{g}(n)$ of ACLMS accelerates the process from under-modelling to over-modelling. It requires a smaller order and less memory in prediction and performs better with a lower minimum achievable MSPE for non-circular data.

(c) Figure 36 compares the two cases of unbalance, namely magnitude and phase distortions. A balanced system has a circular distribution with $\rho = 0$, while an unbalanced system may have any shape except perfect circle. Also, ρ grows as the phase or magnitude difference increases, and the distribution is less similar to a circle. Therefore, as long as the circularity diagram is not a perfect circle, there exist a fault *i.e. unbalance* in the system, and the severity can be determined by the degree of distortion.

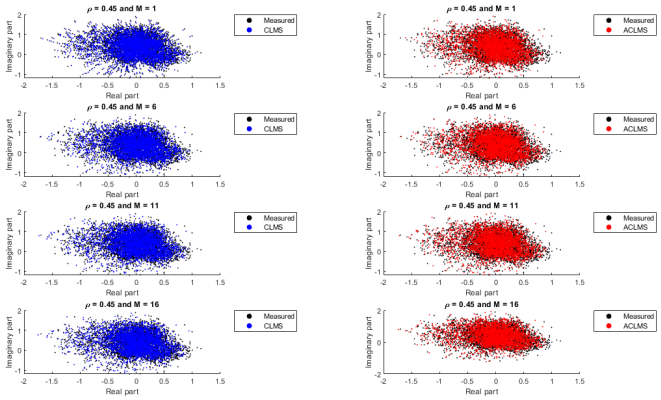

Figure 33: Scatter diagram of medium wind regime

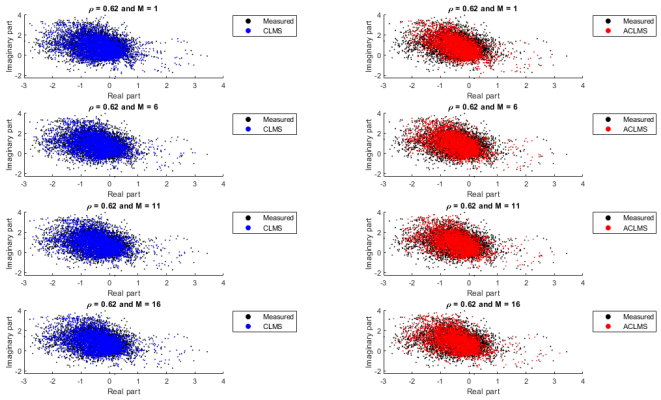

Figure 34: Scatter diagram of high wind regime

(d) Consider a strictly linear autoregressive model with order 1:

$$v(n+1) = h^*(n)v(n) \tag{3.1}$$

The complex Clarke voltage of balanced systems at sample $n+1$ writes:

$$v(n+1) \triangleq v_\alpha(n+1) + jv_\beta(n+1) = \sqrt{\frac{3}{2}} V e^{j\left(2\pi \frac{f_o}{f_s}(n+1)+\phi\right)} \tag{3.2}$$

$$= \sqrt{\frac{3}{2}} V e^{j\left(2\pi \frac{f_o}{f_s} n+\phi\right)} e^{j\left(2\pi \frac{f_o}{f_s}\right)} \tag{3.3}$$

$$= v(n) e^{j\left(2\pi \frac{f_o}{f_s}\right)} \tag{3.4}$$

Compare with Equation 3.1, we have:

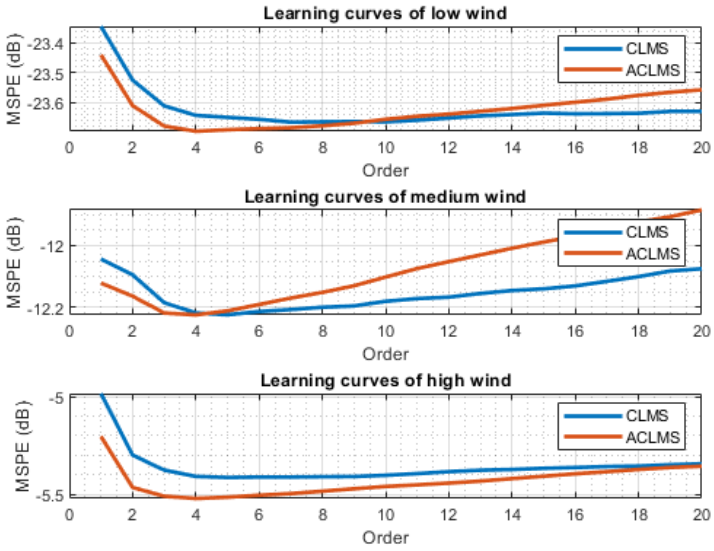

Figure 35: Learning curves of CLMS and ACLMS

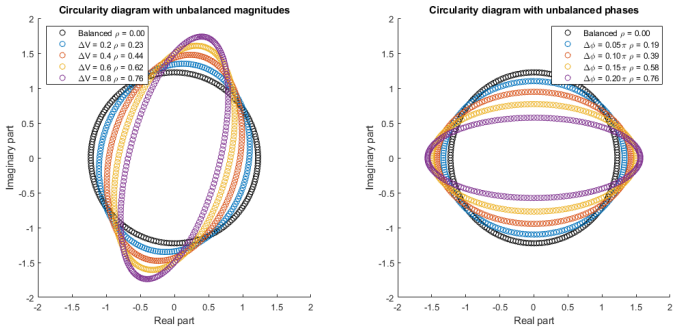

Figure 36: Circularity diagrams of unbalanced complex voltages

$$\begin{aligned} h^*(n) &= e^{j\left(2\pi\frac{f_o}{f_s}\right)} \\ \Rightarrow h(n) &= e^{-j\left(2\pi\frac{f_o}{f_s}\right)} \\ \Rightarrow |h(n)|e^{j\angle h(n)} &= e^{-j\left(2\pi\frac{f_o}{f_s}\right)} \end{aligned} \tag{3.5}$$

Therefore, the real and imaginary parts of both sides equal:

$$\begin{cases} |h(n)| = 1 \\ j\angle h(n) = -j\left(2\pi\dfrac{f_o}{f_s}\right) \end{cases} \tag{3.6}$$

The equation of angles writes:

$$\arctan\left\{\frac{\mathfrak{Im}\{h(n)\}}{\mathfrak{Re}\{h(n)\}}\right\} = -2\pi\frac{f_o}{f_s} \tag{3.7}$$

It proves that the coefficients $h(n)$ determine the frequency of the balanced complex voltage by:

$$f_o(n) = -\frac{f_s}{2\pi}\arctan\left\{\frac{\mathfrak{Im}\{h(n)\}}{\mathfrak{Re}\{h(n)\}}\right\} \tag{3.8}$$

For the unbalanced system, assume the change of magnitude between adjacent samples are negligible *(i.e.* $A(n) = A(n+1), B(n) = B(n+1)$*)*, the complex Clarke voltage at sample $n+1$ can be expressed as:

$$v(n+1) = A(n+1)e^{j\left(2\pi\frac{f_o}{f_s}(n+1)+\phi\right)} + B(n+1)e^{-j\left(2\pi\frac{f_o}{f_s}(n+1)+\phi\right)} \tag{3.9}$$

$$\approx A(n)e^{j\left(2\pi\frac{f_o}{f_s}n+\phi\right)}e^{j\left(2\pi\frac{f_o}{f_s}\right)} + B(n)e^{-j\left(2\pi\frac{f_o}{f_s}n+\phi\right)}e^{-j\left(2\pi\frac{f_o}{f_s}\right)} \tag{3.10}$$

On the other hand, start from the widely linear equation:

$$v(n+1) = h^*(n)v(n) + g^*(n)v^*(n) \tag{3.11}$$

$$= h^*(n)\left(A(n)e^{j\left(2\pi\frac{f_o}{f_s}n+\phi\right)} + B(n)e^{-j\left(2\pi\frac{f_o}{f_s}n+\phi\right)}\right) \tag{3.12}$$

$$+ g^*(n)\left(A^*(n)e^{-j\left(2\pi\frac{f_o}{f_s}n+\phi\right)} + B^*(n)e^{j\left(2\pi\frac{f_o}{f_s}n+\phi\right)}\right) \tag{3.13}$$

Equating 3.9 and 3.11, we have:

$$\begin{cases} A(n)e^{j\left(2\pi\frac{f_o}{f_s}\right)} = h^*(n)A(n) + g^*(n)B^*(n) \\ B(n)e^{-j\left(2\pi\frac{f_o}{f_s}\right)} = h^*(n)B(n) + g^*(n)A^*(n) \end{cases} \tag{3.14}$$

Therefore,

$$\begin{cases} e^{j\left(2\pi\frac{f_o}{f_s}\right)} = h^*(n) + g^*(n)\dfrac{B^*(n)}{A(n)} \\ e^{-j\left(2\pi\frac{f_o}{f_s}\right)} = h^*(n) + g^*(n)\dfrac{A^*(n)}{B(n)} \end{cases} \tag{3.15}$$

Notice both equations compose a complex conjugate pair. Hence,

$$h^*(n) + g^*(n)\frac{B^*(n)}{A(n)} = h(n) + g(n)\frac{A(n)}{B^*(n)} \tag{3.16}$$

Equation 3.16 is essentially a quadratic equation and can be rewritten as:

$$g^*(n)\left(\frac{B^*(n)}{A(n)}\right)^2 + (h^*(n) - h(n))\left(\frac{B^*(n)}{A(n)}\right) - g(n) = 0 \tag{3.17}$$

whose solutions are given by:

$$\frac{B^*(n)}{A(n)} = \frac{-(h^*(n)-h(n)) \pm \sqrt{(h^*(n)-h(n))^2 + 4g^*(n)g(n)}}{2g^*(n)} \tag{3.18}$$

$$= j\frac{\mathrm{Im}\{h(n)\} \pm \sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2}}{g^*(n)} \tag{3.19}$$

Substituting Equation 3.15 with 3.18 leads to:

$$e^{j\left(2\pi\frac{f_o}{f_s}\right)} = h^*(n) + j\left(\mathrm{Im}\{h(n)\} \pm \sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2}\right) \tag{3.20}$$

$$= \mathfrak{Re}\{h(n)\} \pm j\sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2} \tag{3.21}$$

The eqality on angle gives:

$$2\pi\frac{f_o}{f_s} = \arctan\left(\frac{\pm\sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2}}{\mathfrak{Re}\{h(n)\}}\right) \tag{3.22}$$

which demonstrates the relationship between the frequency of the unbalanced voltage and coefficients $h(n)$ and $g(n)$:

$$f_o = \frac{f_s}{2\pi}\arctan\left(\frac{\pm\sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2}}{\mathfrak{Re}\{h(n)\}}\right) \tag{3.23}$$

The positive expression is used according to the physical definition of frequency.

(e) We applied CLMS and ACLMS on the balanced and unbalanced power systems. Phase deviations of $\Delta_b = 0.2\pi, \Delta_c = 0.4\pi$ and magnitude deviations of $\Delta V_a = -0.4, \Delta V_c = 0.4$ are introduced for unbalance. Figure 37 indicates that both algorithms detected the 50 Hz component for a balanced system, although there is a large overshoot for ACLMS in the transient state. One possible reason is that when the sample size is small, there can be over-fitting and it is hard for ACLMS with 2 degrees of freedom to determine whether the pattern is elliptical or circular. In comparison, CLMS assumes the data to be circular that coincides with the balanced model, and the result converges to the true value after 10 trials. Nevertheless, for the unbalanced systems with the fault in phase or magnitude, ACLMS always converges to 50 Hz within 60 steps. In comparison, CLMS ensures fast convergence with less than 20 steps, but the estimate is inaccurate especially when ρ is large. It is as expected since CLMS with 1 degree of freedom predicts based on circular distribution. In other words, it approximates the ellipse with the closest circle, and the estimation error increases as the non-circularity grows. In contrast, as an extension of CLMS with another degree of freedom, ACLMS can describe the unbalance *(i.e. non-circularity)* of any complex-value data with the augmented weight $\mathbf{g}(n)$. Figure 38 also illustrates the advantage in error magnitude for non-circular system. To conclude, balanced systems prefer CLMS for low complexity and fast convergence, while unbalanced system requires ACLMS that can estimate any circular or non-circular complex-value data with sufficiently small error.

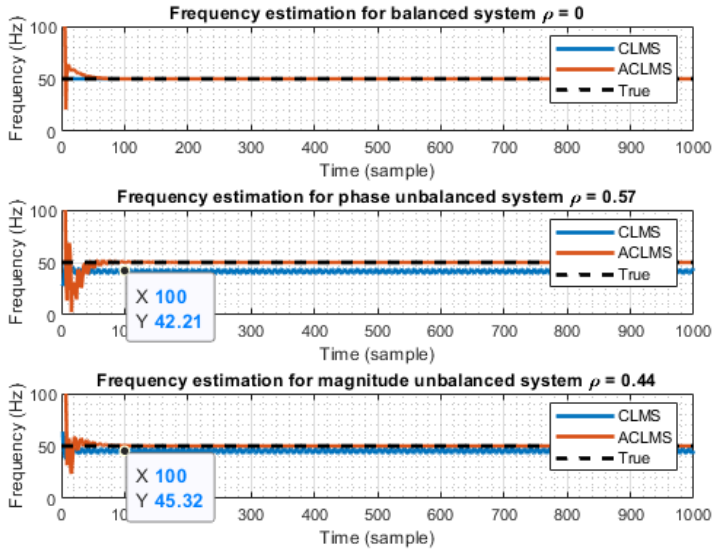

Figure 37: Frequency estimation for balanced and unbalanced systems

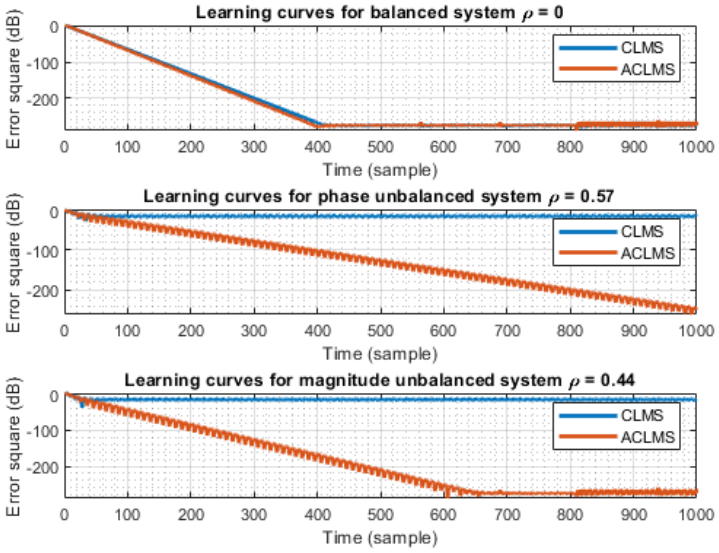

Figure 38: Learning curves for balanced and unbalanced systems

3.2 Adaptive AR Model Based Time-Frequency Estimation

(a) Figure 39 illustrates the frequency and phase of the frequency modulated (FM) signal. A signal is stationary if the properties of the process that generates the signal are unchanged over time. As a special case of chirp, the FM signal is non-stationary because the event-to-event probability is not fixed all the time. However, the `aryule` function assumes the signal as stationary and return the AR coefficient. It can be observed that there is only a single peak around 180 Hz in the overall AR (1) estimate, and the prediction does not capture the frequency change over time. The reason is that the block-based estimate of the AR coefficients is inaccurate for non-stationary signal. The result also shows that when analysing the segments individually, the constant frequency component $f_1(n) = 100$

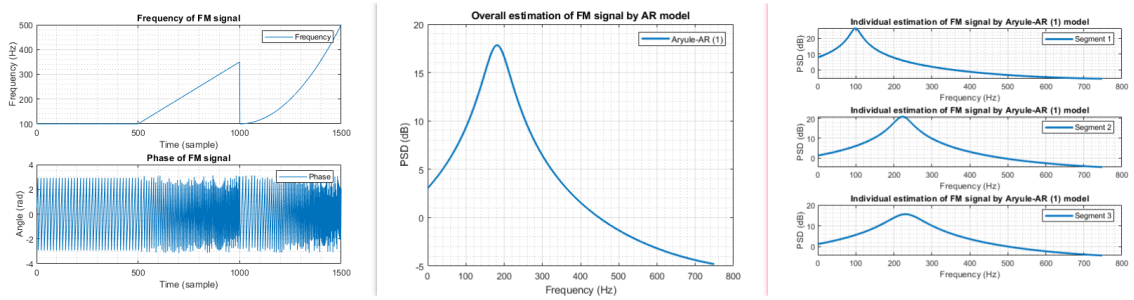

Figure 39: FM signal and Aryule-AR (1) estimates

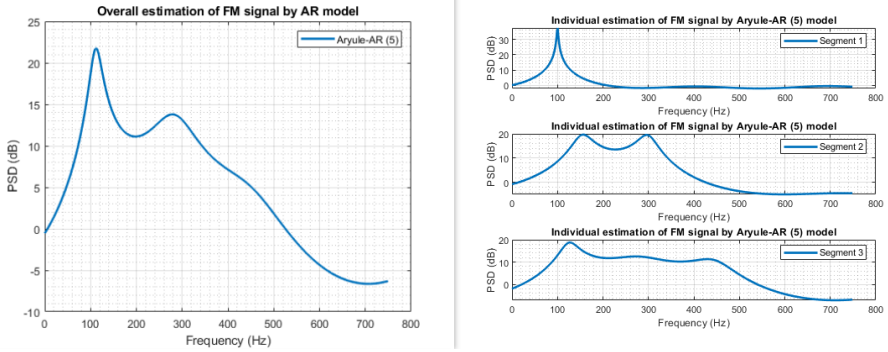

Figure 40: FM signal and Aryule-AR (5) estimates

Hz of the first stage can be modelled, but the Yule-Walker AR (1) estimate failed to detect the changes in the frequency of the second and third segments. It is because the first segment is stationary with a fixed frequency, but the rests are non-stationary. Figure 40 proves that increasing model order does not help to capture the frequency variation, as the estimate is still based on the stationary pattern.

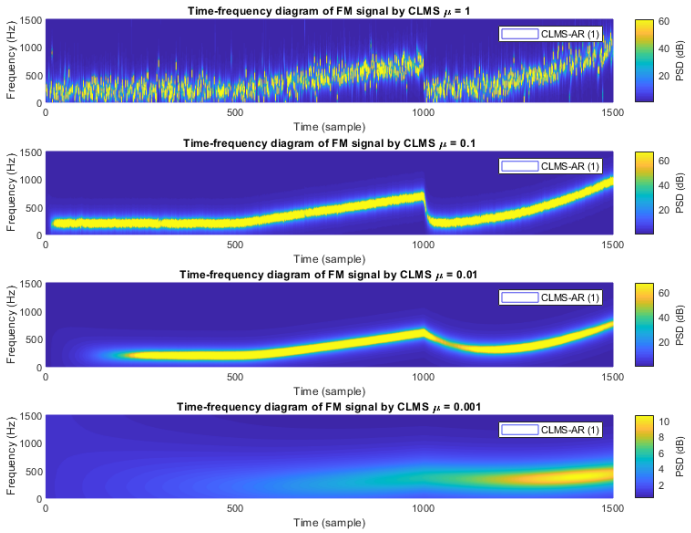

Figure 41: Time-frequency spectrum by CLMS-AR (1) estimates

(b) The time-frequency spectrum by CLMS algorithm that estimates the AR coefficient iteratively is shown in Figure 41. It demonstrates that the adaptive algorithm outperforms the block-based one

for the estimation of non-stationary signals, as the AR coefficient updates over time. Also, there is a trade-off between convergence speed and steady-state variance, and using adaptive step size may help to optimise performance.

3.3 A Real Time Spectrum Analyser Using Least Mean Square

(a) Estimation $\hat{\mathbf{y}}$ as a linear combination of N harmonically related sinusoids is given by:

$$\hat{\mathbf{y}} = \mathbf{F}\mathbf{w} \tag{3.24}$$

where $\mathbf{w}$ is the weight vector and $\mathbf{F}$ is the inverse DFT (IDFT) matrix:

$$\mathbf{F} = \left[\mathbf{\Phi}^0, \mathbf{\Phi}^1, \cdots \mathbf{\Phi}^{N-1}\right] \tag{3.25}$$

with $\mathbf{\Phi} = \left[\phi^0, \phi^1, \phi^2, \cdots, \phi^{N-1}\right]^T$ and $\phi = e^{j\frac{2\pi}{N}}$.

LS solution aims to minimise the cost function defined as the *L-2* norm of the prediction error:

$$\begin{aligned}
\min_{\mathbf{w}} J(\mathbf{w}) &= \min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}\|^2 && (3.26)\\
&= \min_{\mathbf{w}} \left\{(\mathbf{y} - \mathbf{F}\mathbf{w})^H (\mathbf{y} - \mathbf{F}\mathbf{w})\right\} && (3.27)\\
&= \min_{\mathbf{w}} \left\{\mathbf{y}^H\mathbf{y} - \mathbf{w}^H\mathbf{F}^H\mathbf{y} - \mathbf{y}^H\mathbf{F}\mathbf{w} + \mathbf{w}^H\mathbf{F}^H\mathbf{F}\mathbf{w}\right\} && (3.28)
\end{aligned}$$

Take derivative on both sides w.r.t. $\mathbf{w}$:

$$\begin{aligned}
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\left(\mathbf{y}^H\mathbf{y} - \mathbf{w}^H\mathbf{F}^H\mathbf{y} - \mathbf{y}^H\mathbf{F}\mathbf{w} + \mathbf{w}^H\mathbf{F}^H\mathbf{F}\mathbf{w}\right) && (3.29)\\
&= -2\mathbf{F}^H\mathbf{y} + \mathbf{F}^H\mathbf{F}\mathbf{w} && (3.30)\\
& && (3.31)
\end{aligned}$$

The optimum weight $\mathbf{w}^\star$ that minimises the cost function occurs when the derivative is 0, leading to:

$$\mathbf{F}^H\mathbf{F}\mathbf{w}^\star = 2\mathbf{F}^H\mathbf{y} \tag{3.32}$$

If $\mathbf{F}^H\mathbf{F}$ is full-rank, the optimum weight $\mathbf{w}^\star$ writes

$$\mathbf{w}^\star = \left(\mathbf{F}^H\mathbf{F}\right)^{-1}\mathbf{F}^H\mathbf{y} \tag{3.33}$$

Since IDFT matrix $\mathbf{F}$ is unitary, $\left(\mathbf{F}^H\mathbf{F}\right)^{-1} = \mathbf{I}$ and the equation boils down to:

$$\mathbf{y} = \left(\mathbf{F}^H\right)^{-1}\mathbf{w}^\star = \mathbf{F}\mathbf{w}^\star \tag{3.34}$$

which is the IDFT formula.

In conclusion, IDFT can be expressed by a full-rank matrix $\mathbf{F}$ containing N harmonically related sinusoids. The optimum weight $\mathbf{w}^\star = \mathbf{F}^H\mathbf{y} = \mathbf{F}^{-1}\mathbf{y}$ is the Fourier coefficient obtained by DFT, and IDFT on $\mathbf{w}$ gives an optimal linear estimation of the original signal regarding mean error square.

(b) As mentioned above, the coefficients of the optimum linear approximation of the original signal can be obtained by DFT. With a unitary matrix, DFT maps a signal of N samples to another representation of N frequencies by a linear projection, using N harmonically related sinusoids with frequencies being a multiple of f_s/N. It can be interpreted as "change of basis" from the time domain where the basis is L-2 to frequency domain where the kernels are sinusoids.

The projection involves linear combination and basis selection, and harmonically related sinusoids are used in DFT. However, if the sampling frequency f_s is high, or only a few samples are available, both time and frequency resolution will be relatively low because DFT is a one-to-one mapping. It demonstrates that signal processing does not provide any extra information, and DFT only enables another perspective on the observation. In other words, as a linear approximation, the projection onto the frequency subspace spanned by harmonic sinusoids can only maintain or reduce the mutual information.

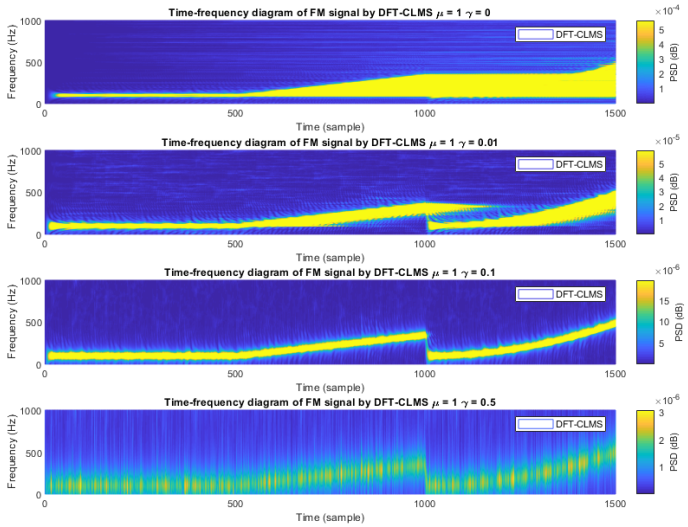

Figure 42: Time-frequency spectrum of FM signal by DFT-CLMS estimates

(c) Figure 42 illustrates the time-frequency diagram of the FM signal by DFT-CLMS algorithm. The case of $\gamma = 0$ successfully estimated the stationary part with constant frequency in the first segment. Nevertheless, it only captures the envelope of the desired frequency spectrum for non-stationary signals. There appears to be a lasting effect that DFT-CLMS with $\gamma = 0$ tends to remember the components detected and regard them as remaining in the rest of the estimation. The reason is that the weight is updated adaptively rather than calculated directly at each time instant, and the new frequency components dominate the error variation in gradient descent. Therefore, the Fourier coefficients superimposed over time, as there is no effective error backpropagation mechanism. Even if we pad zeros to a length of 4096, the everlasting effect remains as shown by Figure 43. Also, for the FM signal with length 1500, the weight of DFT-CLMS is 1500 dimensional, and the curse of dimensionality leads to a slower weight update.

One possible solution to accelerate error back propagation is to introduce a leakage $\gamma \neq 0$ to reduce the dependency in the iterations. Figure 42 demonstrates its advantage over standard DFT-CLMS. A proper γ as 0.1 can achieve a balance between slow error backpropagation and large estimation

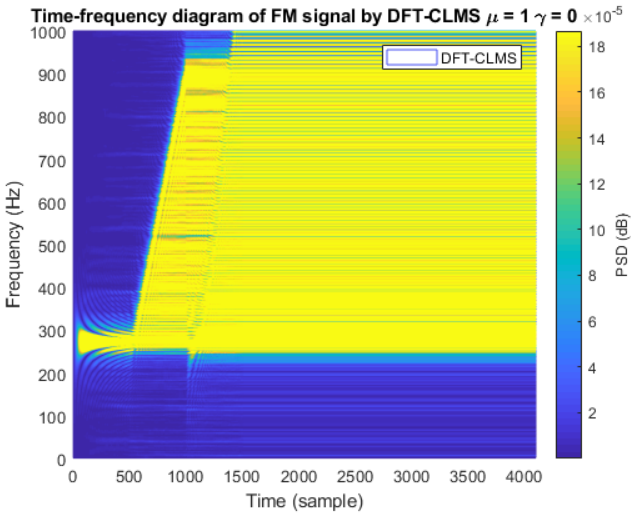

Figure 43: Time-frequency spectrum of zero-padded FM signal by DFT-CLMS estimates

variance. It provides more innovations in each update and relies less on the weight at the previous instant, and the algorithm performs better than the standard DFT-CLMS for non-stationary signals.

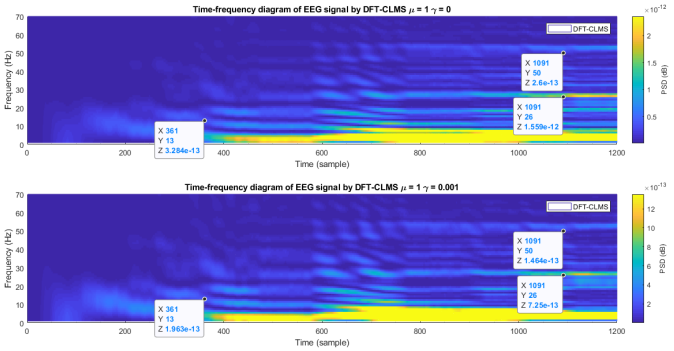

Figure 44: Time-frequency spectrum of EEG data by DFT-CLMS estimates

(d) The time-frequency diagrams of the standard and leaky DFT-CLMS are shown in Figure 44. Note the strong response at 8–10 Hz is alpha rhythm due to the tiredness of the subject. The SSEVP frequency at 13 Hz is detected within 400 steps, and as the algorithm iterates, not only the accuracy increases but also the harmonic at 26 Hz and the power frequency at 50 Hz can be observed. Nevertheless, the leaky algorithm leads to a more ambiguous result, as it tends to drop out some dependency and rely on more recent samples. It is helpful for non-stationary signals that require fast error back propagation, but the leakage reduces the convergence speed and model stability for stationary signals because information is lost. Moreover, the resolution of the plot can be improved by increasing the sample length, but the computational cost grows as well.

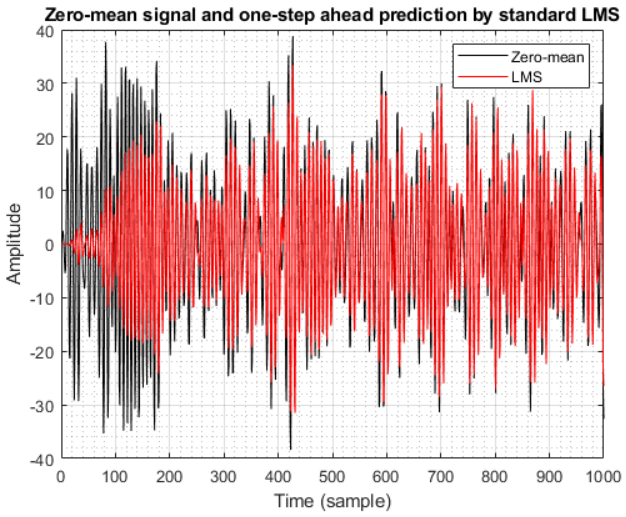

Figure 45: Zero-mean signal and one-step ahead prediction by standard LMS

4 From LMS to Deep Learning

4.1 Linear Prediction by LMS

Figure 45 illustrates the zero-mean signal and its one-step-ahead prediction by standard LMS based on linear AR (4) model. It takes around 180 steps for the estimate to converge with a learning rate of $\mu = 1 \times 10^{-5}$, and the result captures the variation trend of the time series despite a conservative prediction with an absolute value smaller than the truth in most cases. The linear estimate provides a large MSE of 16.03 dB with the prediction gain $R_p = 5.20$ dB, and a more accurate model is desired.

4.2 Nonlinearity by Tanh

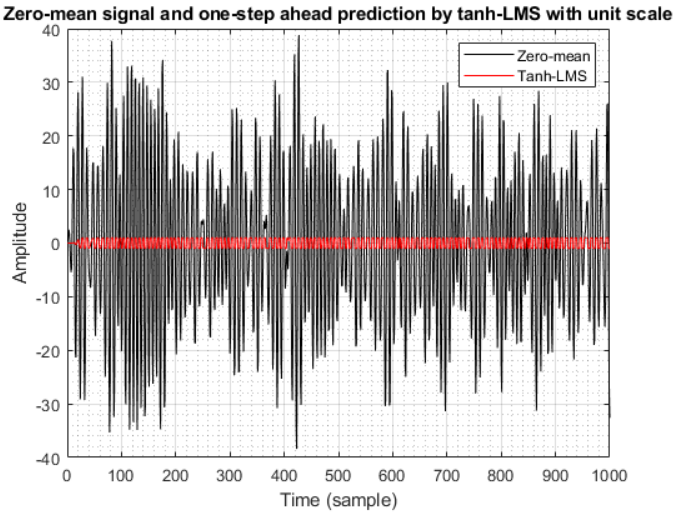

Figure 46: Zero-mean signal and one-step ahead prediction by tanh-LMS with unit scale

A non-linear activation function can be added to the output of the LMS to model the non-linearity

of the real-world data. Figure 46 shows an prediction instance by a dynamic perceptron model with `tanh` activation function of unit scale. `tanh` maps an input of $(-\infty, \infty)$ to a output within $(-1, 1)$. Although the estimate by tanh-LMS successfully captures the frequency variation of the time series, it does not reflect the change in amplitude as the magnitude is limited. In this case, the MSE is 22.94 dB, and the prediction gain is $R_p = -23.19$ dB, being worse than the linear estimate. Therefore, `tanh` with unit scale is not an appropriate activation function.

4.3 Nonlinearity by Scaled Tanh

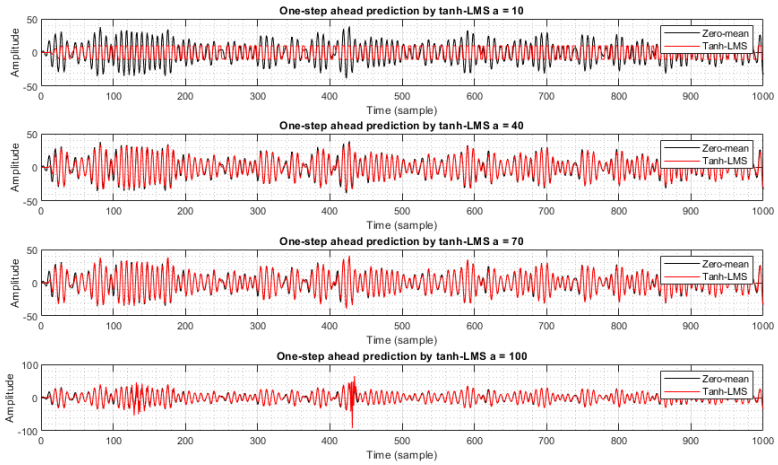

Figure 47: Zero-mean signal and one-step ahead prediction by tanh-LMS with different scales

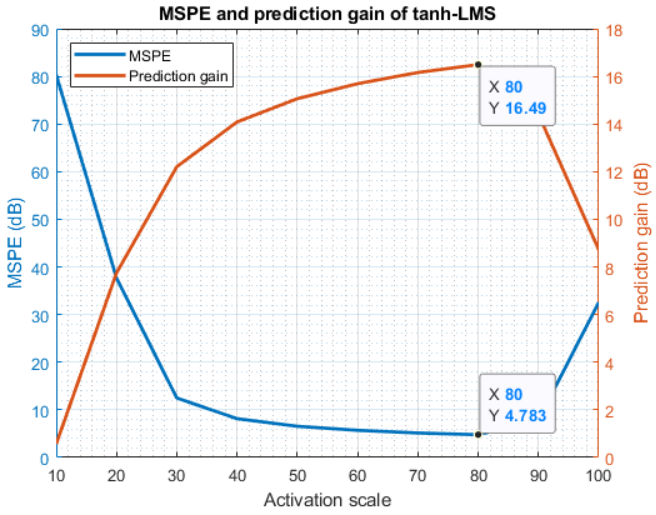

Figure 48: MSPE and prediction gain of tanh-LMS

Figure 47 illustrates the relationship between the activation scale a and the predicted signal. If a is too small, the available maximum magnitude of the model is not large enough to approximate the original signal, and the prediction appears truncated in those regions. As a increases, the available range of the non-linear model becomes wider, but overfitting and instability may occur if the scale is too large. For instance, although the model of $a = 100$ results in a smaller deviation in most times,

strong oscillation happens when the signal amplitude or frequency change significantly. Larger a that amplifies the non-linearity also leads to faster convergence.

The MSPE and prediction gain of the non-linear tanh-LMS is shown in Figure 48. It suggests that $a \in [30, 90]$ is appropriate and the optimum scale $a^\star = 80$ for the time series. Compared with the linear estimation by standard LMS, the minimum $MSPE = 4.783$ dB is significantly smaller while the maximum prediction gain $R_p = 16.49$ dB is 11 dB larger. Therefore, the tanh-LMS outperforms the standard LMS in the estimate of non-linear signals.

4.4 Bias by Augmented Input

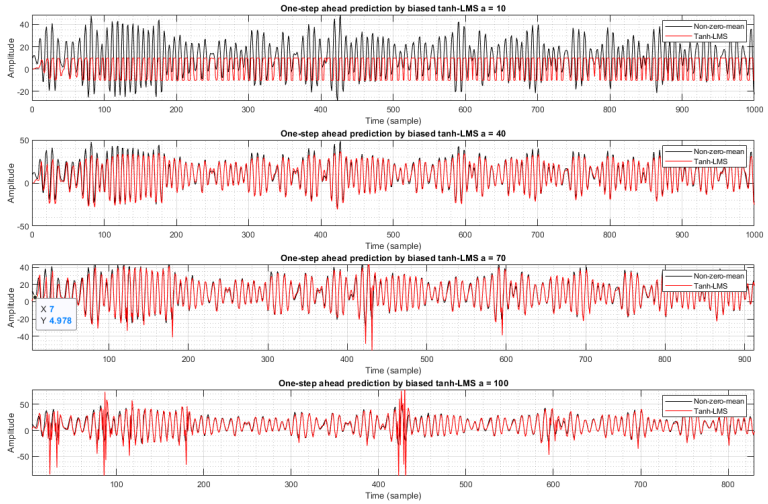

Figure 49: Non-zero-mean signal and one-step ahead prediction by biased tanh-LMS with different scales

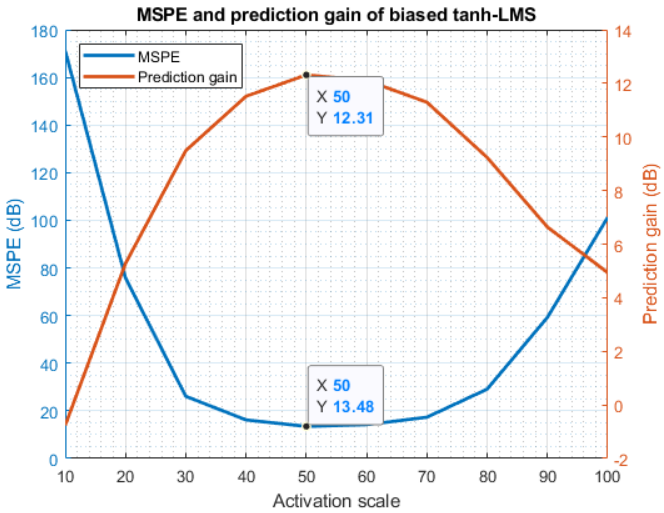

Figure 50: MSPE and prediction gain of biased tanh-LMS

Figure 49 compares the original non-zero-mean signal and the estimate by tanh-LMS model with bias. It is observed that by using an additional constant input row, the algorithm managed to shift the prediction vertically to capture the mean adaptively. Also, the compensation is related to the

scale, and the adjustment by a small a may be insufficient to shift the prediction back to the average level. The reason is that the prediction of mean is also influenced by the activation function by $\phi(\mathbf{w}^T\mathbf{x} + b)$. As a increases, the magnitude restriction by `tanh` reduces, and the gap between the average level of prediction and original signal narrows. As for the convergence speed, for a fixed $a = 70$, the unbiased tanh-LMS converges after 20 steps while the biased one converges within 15 trials. One possible reason is that the non-zero mean has a positive contribution to the update of weight in the beginning. In other words, both zero-mean and non-zero-mean signals may change fast in the beginning, but the prediction with bias tends to capture the influence of both mean and signal variation. For instance, the estimate without bias oscillates around zero in the first few predictions with a large error. In comparison, the estimate with bias has a trend of increasing due to the positive mean. Interestingly, the variation of non-zero-mean signal results in a fast intersection of prediction and truth (as indicated by the point in Figure 49), leading to smaller estimation error. However, the prediction of the zero-mean signal needs to catch up with the change, and it takes a longer time to converge.

Figure 50 reveals a minimum MSPE of 13.48 dB and a maximum prediction gain of 12.31 dB, corresponding to an optimum scale $a = 50$. The error of biased prediction is larger than the one without bias, due to the estimation error of the mean as the curse of an additional dimension.

4.5 Weight Pretraining

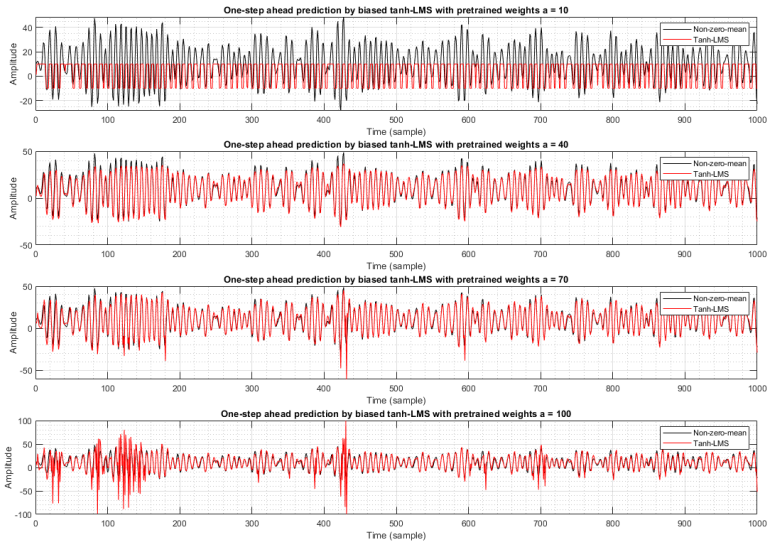

Figure 51: Non-zero-mean signal and one-step ahead prediction by pretrained biased tanh-LMS with different scales

The first 20 samples of the signal are used to pretrain the weight with 100 iterations for fast convergence. Figure 51 shows that the prediction overlaps with the truth almost perfectly. For a proper scale as $a = 40$, it converges even faster in less than 5 steps. It demonstrates the benefit of overfitting the first few samples on convergence speed. When the activation scale a is large, there can be more overshoots than the standard model, as the pretraining overfits the first minibatch. Figure 52 proves its advantage of 3.7 dB MSPE with 34% larger prediction gain in the optimum case. To conclude, pretraining the weight by overfitting a small number of samples can guarantee fast convergence, low error and large prediction gain, but the batch size should be determined carefully to balance the generality and complexity.

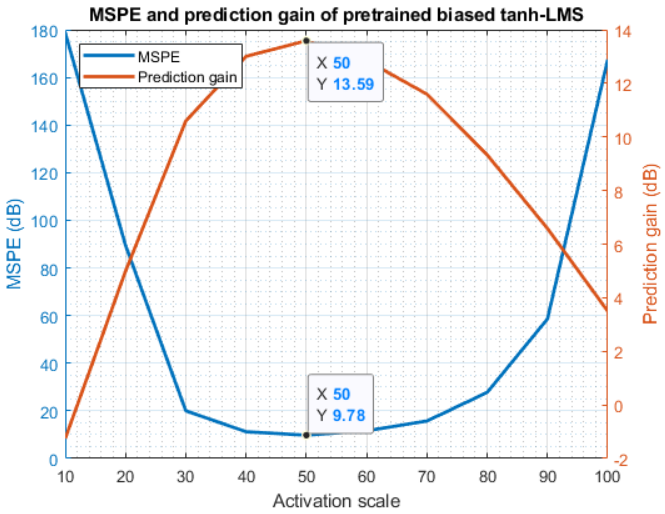

Figure 52: MSPE and prediction gain of pretrained biased tanh-LMS

4.6 Error Backpropagation

Backpropagation is used in supervised learning to calculate the gradient of descent for weight modifications. The error is obtained at the output, and its influence propagates backwards through layers for parameter adjustment. The target of backpropagation is to determine the value that weights and biases should change to reduce the cost function most efficiently, and the problem is equivalent to calculate the negative gradient of the cost function w.r.t. weights and biases.

Backpropagation can be used to train a deep network, and there are mainly five steps in the training process. First, the weights and biases are initialised to random values, as parameters will converge with the help of error backpropagation. The second step is to check the performance of the existing network. The labelled input is passed to the network, and the model output is examined. Third, a cost function is used to measure the difference between the output and its expectation. Fourth, the partial derivatives of the cost function w.r.t. parameters are determined, and the gradient of one layer can be reused to calculate that of the previous layer. Finally, the error is backpropagated, and the weights and biases are updated accordingly. Step 2 to 5 are repeated until the network converges with a small error.

Compared with simple activation functions as `tanh`, the deep network model with hidden layers are more descriptive for non-linearity problems, and the two can be combined for more expressive models.

4.7 Performance of Deep Network

The desired non-linear signal is a combination of sinusoids with different amplitude and frequency by a highly non-linear activation function $\phi(n)$, as illustrated by Figure 53. To fit the signal, the deep network is trained with $L = 4$ hidden layers, 2×10^4 epochs, learning rate $\mu = 0.01$ and noise variance $\sigma_n^2 = 0.05$. Figure 54 compares the estimates by single linear neuron, single tanh neuron, and deep network. Although successfully captured the general trend of the training set, both single neuron models result in large errors when fitting the testing curve. The main reason is that the degree of freedom provided by a single neuron is insufficient to model the high non-linearity of the activation function. Despite `tanh` enables non-linear mapping from input to output, there is a highly linear region around the origin that the function can be approximated as $y = x$ that restricts the

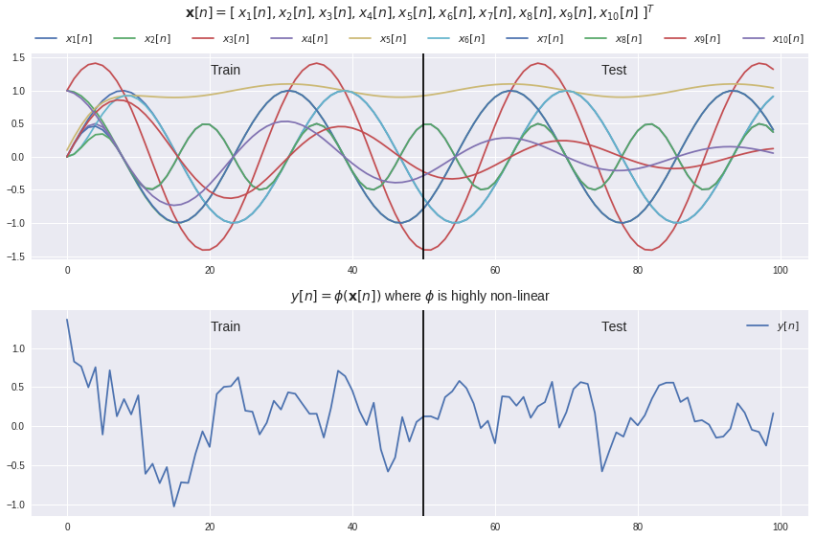

Figure 53: Linear input and non-linear output signals

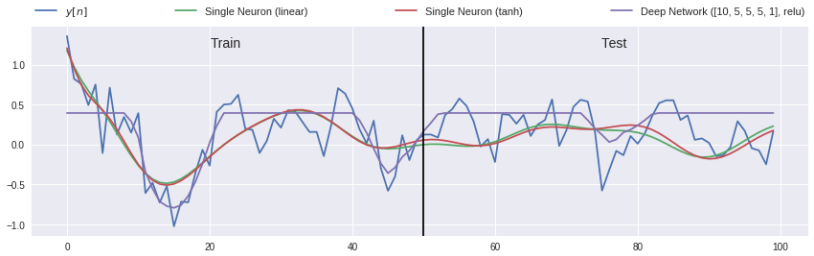

Figure 54: Non-linear signal and estimates by linear single neuron, tanh single neuron, and deep network with default parameters

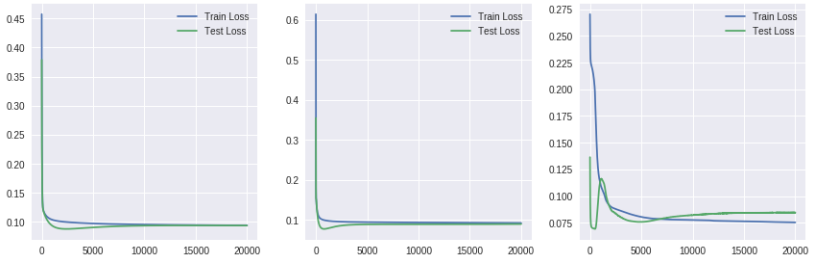

Figure 55: Learning curves by linear single neuron, tanh single neuron, and deep network with default parameters

ability in fitting non-linearity. Therefore, the improvement of `tanh` activation function denoted by the difference between the red and green curves is not obvious. Also, the prediction has a strong dependency on the input. The weight of the single neuron models depends on the Fourier series, and the combination cannot model any sudden change according to the Gibbs phenomenon. Moreover, it is susceptible to the trend of input because the number of input is limited. For instance, nearly all kernels are strictly concave in sample 20–40, and the estimation is also concave and failed to describe the convex-concave behaviour of the desired signal.

In comparison, the estimate of the deep network is maintained at the local mean and ignores the small-scale variations of the signal. The gap in the testing set is larger than the training set. Figure 55 proves that the minimum prediction error by the deep network is 0.075, smaller than those of

dynamical perceptrons. It is because more layers provide a higher degree of freedom in non-linearity description, and guarantee a more expressive and accurate model with a smaller error. However, it takes around 10,000 epochs to converge, slower than the single neuron models. The reason is that the weight is updated by error backpropagation, which can be slow due to the curse of dimensionality and the randomness in stochastic gradient descent (SGD). The learning curve of the deep network on the testing data experiences an oscillation between epoch 0 and 1,500. One possible reason is that the stochastic gradient descent results in an unstable model at the beginning that happens to fit the testing data. As training continues, the network overfits the training set and the test loss increases. In comparison, the learning curves of dynamical perceptions on both training and testing data converge to 0.1 and end up flat, as they are not expressive enough to model the highly non-linear problems, and the weights converge to a pseudo-optimum point in a smaller dimension of signal subspace. Compared with the linear single neuron model that converges after 7,500 updates, `tanh` can compensate some non-linearity to ensure faster convergence within 5,000 steps.

4.8 Different Networks and Comments

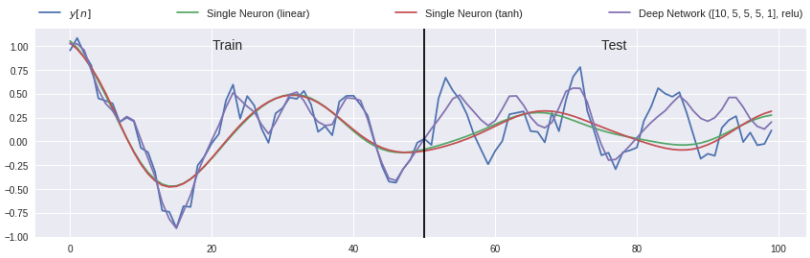

Figure 56: Non-linear signal and estimates by linear single neuron, tanh single neuron, and deep network with noise power $\sigma_n^2 = 0.01$

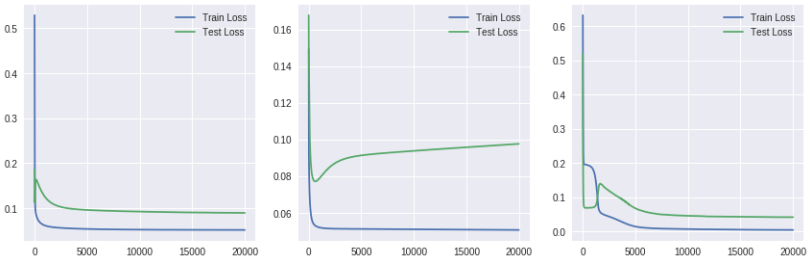

Figure 57: Learning curves by linear single neuron, tanh single neuron, and deep network with noise power $\sigma_n^2 = 0.01$

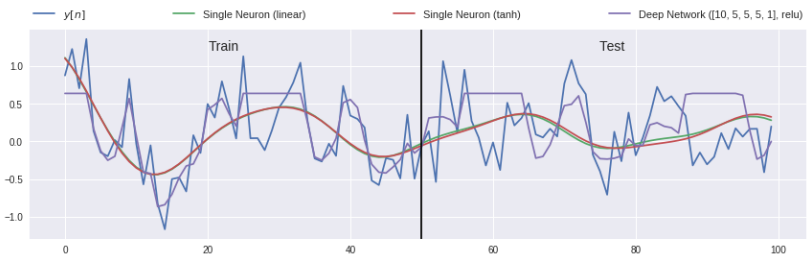

Figure 58: Non-linear signal and estimates by linear single neuron, tanh single neuron, and deep network with noise power $\sigma_n^2 = 0.1$

Figure 56 and 57 show the estimates and learning curves corresponding to a noise power $\sigma_n^2 = 0.01$. As noise decreases, the deep network provides smaller errors for both training and testing data, but

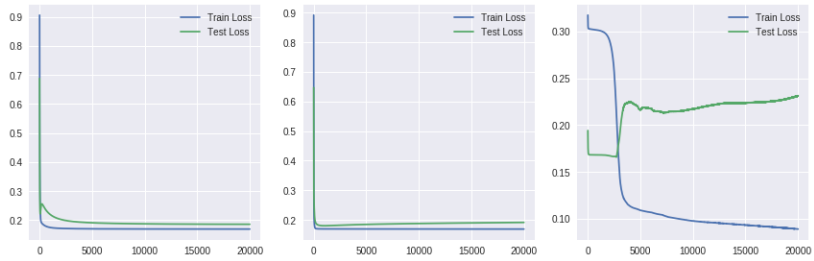

Figure 59: Learning curves by linear single neuron, tanh single neuron, and deep network with noise power $\sigma_n^2 = 0.1$

the performance of dynamical perceptrons only improves on the training set. It is because the single neuron estimators are not expressive enough to capture the non-linearity pattern behind the data. In other words, the optimisation in the training process comes from overfitting the dataset, and the model based on the linear or weak non-linear architecture is not suitable for more generalised non-linear problems. In comparison, the test loss of the deep network is reduced to 0.04 after 10,000 epochs, demonstrating the ability to learn a general non-linear pattern from training data. However, for a larger noise power $\sigma_n^2 = 0.1$, the deep network "think too complicated" and overfits the noise. Although its error on the training set is smaller than 0.1, the test loss is larger than dynamical perceptron.

In conclusion, the deep network model with hidden layers is more expressive to model the non-linearity in real-world problems, and it can produce very accurate predictions with fine-tuned parameters. However, the network requires much more nodes and higher computational complexity, and the convergence speed can be slower than dynamical perceptron. The reason is that the weights are adjusted through error backpropagation and suffer the curse of dimensionality and the randomness in the SGD. Also, a complicated model may lead to overfitting especially when the noise power is relatively large. With the cost of complexity, there are more possibilities by the deep neural network.

5 Appendix: MATLAB code

The source code can be retrieved from `https://github.com/SnowzTail/`.

References

[1] D. P. Mandic, "Adaptive signal processing and machine intelligence," January 2019. pages 7, 15

[2] A. A. Shah and D. W. Tufts, "Determination of the dimension of a signal subspace from short data records," **IEEE Transactions on Signal Processing**, vol. 42, no. 9, pp. 2531–2535, 1994. pages 9

[3] E. Fishler and H. V. Poor, "Estimation of the number of sources in unbalanced arrays via information theoretic criteria," **IEEE Transactions on Signal Processing**, vol. 53, no. 9, pp. 3543–3553, 2005. pages 9

[4] D. P. Mandic, "A generalized normalized gradient descent algorithm," **IEEE signal processing letters**, vol. 11, no. 2, pp. 115–118, 2004. pages 20